Ludwig-Maximilians-Universität
München

Institut für Informatik

Fortgeschrittenenpraktikum

Modellierung von Betriebsabläufen mit Unicenter TNG

Vyacheslav Sklarevskiy

Aufgabensteller: Prof. Dr. Heinz-Gerd Hegering
Betreuer: Rainer Hauck
Igor Radisic

Kapitel 1 Einleitung **3**

Kapitel 2 Die Teilbereiche der Enterprise Management Schicht **6**

2.1 Die Architektur von Unicenter TNG **6**

2.2 Die Funktionen für DV-Management **7**

2.2.1 Eventmanagement 8

2.2.2 Distributed State Machine (DSM) 12

2.2.3 Überblick über die anderen Funktionen für DV-Management 14

Kapitel 3 Agent Technology: Architektur und Funktionalität **16**

3.1 Architektur **16**

3.2 Agent Funktionalität **19**

Kapitel 4 Die betriebsablauforientierte Sicht **21**

Kapitel 5 Prototyp eines SNMP-Agenten **27**

5.1 Beschreibung eines Prototyps **27**

5.2 Die SDK- Beschreibung **28**

Kapitel 6 Zusammenfassung **30**

Anhang Troubleshooting **31**

Abbildungsverzeichnis **32**

Literaturverzeichnis **33**

Kapitel 1 Einleitung

Auf dem Weg in die globale Ökonomie investieren Unternehmen sehr viel Geld in die Ausstattung ihrer Informationstechnologie. Aus diesem Grund ist es für ein Unternehmen inakzeptabel, wenn eine Transaktion oder ein Verkaufsabschluß nicht zustande kommt, weil das E-Mail-System nicht funktioniert, eine Festplatte voll oder ein Server ausgefallen ist [wwwUDE].

Je komplizierter und unvorhersehbarer die DV-Umgebung eines Unternehmens wird, desto notwendiger ist es, daß die IT-Ressourcen strukturiert, flexibel und brauchbar sind. Die Verwaltung einer solchen Unternehmens-DV erfordert mehr als eine Reihe von Anwendungen oder intelligenten Managementansätzen.

Um eine an wirtschaftlichen Kriterien orientierte Infrastruktur betreiben zu können, ist ein Überblick über Komponenten, Auslastungsgrad, Perfomance und Angemessenheit für die fachlichen Anforderungen notwendig.

Trotz aller Konsolidierungsbemühungen werden in zahlreichen Unternehmen noch immer recht heterogene Infrastrukturen und Systeme betrieben, die schwer und aufwendig zu administrieren sind. Einen umfassenden Ansatz bietet hierfür **das integrierte Management**, das die Administration durch den Einsatz intelligent gestalteter Werkzeuge weitgehend automatisiert realisiert .
Ein integriertes Management bewerkstelligt eine zentrale Kontrolle über alle Plattformen und Anwendungen hinweg und sichert die Einhaltung einheitlicher Regeln. Ursprünglich getrennte Disziplinen, wie Netz-, System- und Anwendungsmanagement, verschmelzen zu einem integrierten Management. Dies erhöht die Verfügbarkeit und Performance der Anwendungen. Die Administration aller Systeme über ein integriertes Management vereinfacht die Administration und macht diese effizienter. Hersteller wie IBM, Computer Associates, Hewlett Packard und andere bieten hier die entsprechenden Lösungen [RiSt1998].

Eine **Managementplattform** ist ein Werkzeug, das Planung, Konfiguration, Steuerung, Überwachung, Fehlerbehebung sowie Verwaltung von Rechnernetzen, Netzressourcen und verteilten Systemen ermöglicht.

Managementplattformen haben folgende Eigenschaften:
- sie laufen in offenen Systemumgebungen,
- sie bieten eine graphische Oberfläche,
- sie sind nicht auf das Management von Ressourcen bestimmter Hersteller beschränkt,
- sie sind nicht auf Funktionsbereiche beschränkt,
- sie sind modular aufgebaut, wobei einzelne Module auf verschiedene Systeme verteilt werden können,
- sie basieren auf objektorientierten Datenmodellen,
- sie bieten eine auf Standards (Architekturen) beruhende Laufzeitumgebung für verteilte Managementanwendungen und eine entsprechende Entwicklungsumgebung,
- sie bieten offen gelegte Programmierschnittstellen für Anwendungen an,
- sie unterstützen die Integration des Managements in mehrfacher Hinsicht: Die verschiedenen Managementobjekte (Objekte des Netz-, System- und Anwendungsmanagements) können wegen der gemeinsamen darunterliegenden Informationsmodelle vereinheitlicht gesehen werden. Auch eine funktionelle Integration wird unterstützt durch eine vereinheitliche Plattform-Datenbank und gemeinsam benutzte Basisanwendungen.

Abbildung 1.1 zeigt schematisch den funktionalen Aufbau einer Managementplattform [Heg1999].

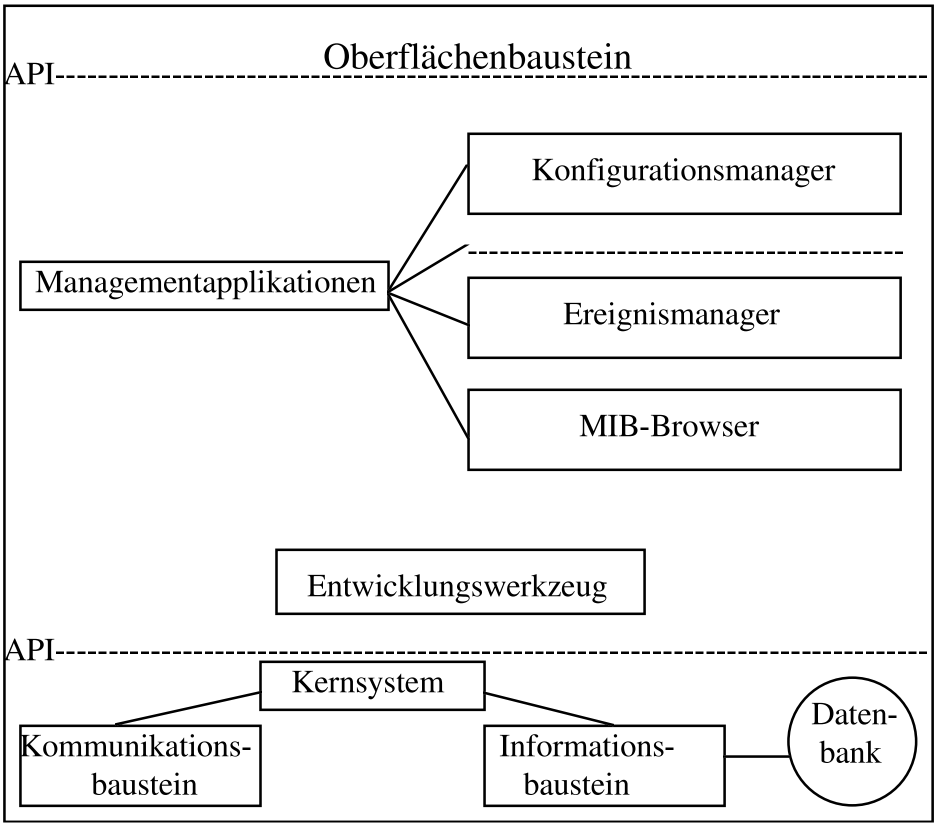


ABBILDUNG 1.1
AUFBAU VON MANAGEMENTPLATTFORM

Das Kernsystem sorgt für Kommunikation und Koordination aller anderen Bausteine. Ein *Kommunikationsbaustein* ermöglicht die Kommunikation mit Fremdsystemen, wobei diese entweder Netzkomponenten, Endsysteme oder Managementstationen sind. Ein *Informationsbaustein* implementiert das objektorientierte Informationsmodell der Managementplattform. Er stellt Dienste bereit für das Kreieren und Verwalten von Managementinformationen. Ein *Oberflächenbaustein* stellt das Netz graphisch dar und ermöglicht dem Benutzer den Zugang zu den Managementanwendungen. Verschiedene *Basisanwendungen*, z.B. Ereignismanagement, MIB-Browser, sind bereits Bestandteil der meisten Managementplattformen. *Entwicklungswerzeuge* erlauben die Erweiterung einer bestehenden Plattform.

Ziel dieses Praktikums ist es, das Betriebsablaufmanagement von Unicenter TNG zu evaluieren und festzustellen, ob es den modernen Anforderungen an die Managementplattformen entspricht.

Der Wunsch nach Betriebsablaufmanagement ist dadurch begründet, daß man nicht eine einzige Ressource, wie z.B. einen Router oder einen Printer, sondern jede vorhandene DV-Ressource, einschließlich Hardware, Datenbanken und Anwendungen, **die an einem Betriebsablauf beteiligt sind**, *unabhängig von deren Typ oder Standort* verwalten will.

Unternehmen können ihre verteilten DV-Umgebungen heutzutage kaum mehr überschauen. Die Anzahl der Systeme, Plattformen und Standorte ist einfach zu hoch. Hinzu kommt der kontinuierliche technische Fortschritt. Dank der *betriebsablauforientierten Sichtweise* lassen sich Unternehmensabläufe mit dem entsprechenden Ressourcenbedarf verbinden und die betroffenen Ressourcen innerhalb eines bestimmten betrieblichen Funktionsbereichs überwachen [RiSt1998].

Bei Auftreten eines Problems kann das Augenmerk auf die zugehörige Ressource gerichtet werden, ohne daß weitere Ressourcen berücksichtigt werden müßten, die anderen Unternehmensbereichen zugeordnet sind.

Die Vorgehensweise des Praktikums:

- Installation von Unicenter TNG und zusätzlicher Komponenten:
  Hier werden Gedanken gemacht, welche Softwarekomponenten im Hinsicht auf die Aufgabenstellung außer der Managementplattform selber zu installieren sind. Anschließend werden Unicenter TNG und zusätzliche Komponenten, nämlich Agent Technology und Event Management auf dem Manager-Host und Unicenter System Agent auf dem zu überwachenden Host installiert.
- Modellierung eines einfachen Beispielbetriebsprozesses:
  Hier wird es überlegt, welche Betriebsablaufkomponente sich überhaupt an einem einfachen Betriebsablauf, hier ein Printer-Betriebsablauf, beteiligen, welche davon man überwachen möchte. Anschließend wird geklärt, in welcher logischen Beziehung die einzelnen Komponenten zueinander stehen (das können „AND"- oder „OR"-Beziehungen sein) und wie der Ausfall der einzelnen Betriebsablaufkomponenten auf den Gesamtzustand des Betriebsablaufs auswirken kann.
- Umsetzung in Unicenter TNG:
  Hier überlegt man sich, wie die Betriebsablaufkomponente, die man gerne überwachen möchte, mittels Unicenter TNG überwacht werden, welche Tools Unicenter standardmäßig dazu bietet, und ob diese Tools ausreichen.
- Erstellung zusätzlicher Softwarekomponenten mittels Unicenter SDK:
  Hier werden Gedanken gemacht, was die Tools leisten sollen, die man für die Überwachung von Betriebsablaufkomponenten braucht und die Unicenter TNG standardmäßig nicht bietet. Anschließend wird die benötigte Softwarekomponente prototypisch erstellt.

In Kapitel 2 werden die Architektur von Unicenter TNG sowie einige wichtige unterstützte Managementfunktionen beschrieben und erklärt.
Ausführlich wird in Kapitel 3 auf die Architektur und Funktionalität von Agent Technology eingegangen.
Kapitel 4 untersucht die betriebsablauforientierten Sicht.
In Kapitel 5 wird die Möglichkeit des Unicenter SDKs untersucht, indem ein SNMP-Agent implementiert wird.
Kapitel 6 enthält eine Zusammenfassung sowie abschließende Bemerkungen zu den Evalueirungsergebnissen

Kapitel 2 Die Teilbereiche der Enterprise Management Schicht

In diesem Kapitel werden die Unicenter-Architektur mit den einzelnen dazugehörigen Schichten und deren Aufgabe sowie die Funktionen für DV-Management - Event Management und Distributed State Machine (DSM) - mit denen ich mich während meines Praktikums beschäftigt habe, beschrieben. Weiter im Überblick werden auch die anderen Management-Funktionen grob dargestellt.

2.1 Die Architektur von Unicenter TNG

Die Aufgabe von Unicenter TNG (The Next Generation) ist es, System und Netzwerkmanagement für mittlere und große Unternehmen durchzuführen [wwwCA]. Es managt dabei heterogene Netzwerke, Datenbanken und Applikationen. Das zentrale Element im Aufbau dieses Produktes ist das **Unicenter TNG Framework**, das kostenlos ist, und bereits einige rudimentäre Anforderungen bewältigt.

Unicenter TNG Framework enthält die folgenden Komponenten:
- Object Repository: ein Speicherplatz für alle verwaltete Ressourcen und deren Eigenschaften und Beziehungen
- Real World Interfaces: eine realitätsgetreue Oberfläche
- Event Management (s. Kapitel 2.2.1)
- Auto Discovery: eine Netzwerkanalyse
- Calendar Management
- Virus detection

Um aber komplexere Aufgaben erledigen zu können, ist es notwendig das eigentliche Unicenter TNG sowie weitere Module (z.B. Agent Technology, Web Management oder Storage Management) zu erwerben und auf diese Weise die Funktionalität des Frameworks zu erweitern.

Unicenter TNG ist in drei Schichten aufgebaut (s. Abbildung 2.1): **die WorldView Schicht**, **die Enterprise Management Schicht** und die **Agent-Schicht**.

Die Aufgabe der Agent-Schicht ist es, Daten über verschiedene Geräte zu sammeln und wiederum Informationen an sie zusenden. Dabei werden zwei Arten von Agenten unterschieden: Intelligent Agents und Neugents (s. Kapitel 3). Während die zuerst genannten für das Sammeln, Filtern und Weiterleiten von Daten über das Netzwerk an höhere Schichten zuständig sind, unterscheiden die Neugents zwischen „normalen“ und auffälligen Verhalten im System und lösen gegebenenfalls, falls beispielsweise eine für die Tageszeit ungewöhnlich hohe Datenmenge über das Netzwerk transportiert wird, Alarm aus.

Es stehen für zahlreiche Systeme und Anwendungen Agents zur Verfügung, wie zum Beispiel für SAP R/3, Sybase und Oracle Datenbanken. Auch ist es möglich eigene Agenten mit Hilfe von C/C++ und Software Development Kit (SDK) zu entwerfen (s. Kapitel 6).

Alle Agenten kommunizieren über SNMP mit der nächst höheren **Enterprise Management Schicht**, die die **Aufgabe** hat, die unterschiedlichen Protokolle zu vereinheitlichen. Diese Ebene besteht aus einer API, die eine Vielzahl von Funktionen behandeln kann wie beispielsweise Erkennung von Geräten und Objekten im Netzwerk, Event Management, Speichermanagement und viele andere.

Eine Ebene darüber liegt die **WorldView Schicht**, die aus dem **Common Object Repository (Core)** und dem **User Interface** besteht. Das Core ist eine objektorientierte Datenbank, in der sich Metainformationen über das zu Grunde liegende System (d.h. Wissen, das die Anordnung der Information beschreibt) und natürlich auch Angaben über die verwaltete Objekte befinden.

Das **User Interface** ist das System mit dem im Core enthaltenen Wissen dargestellt werden kann. Hier gibt es eine herkömmliche zweidimensionale sowie eine 3-D Sichtweise. Da all diese Darstellungen auf denselben Daten im Core beruhen, ist es möglich zwischen ihnen hin und her zu schalten, ohne Inkonsistenz befürchten zu müssen.

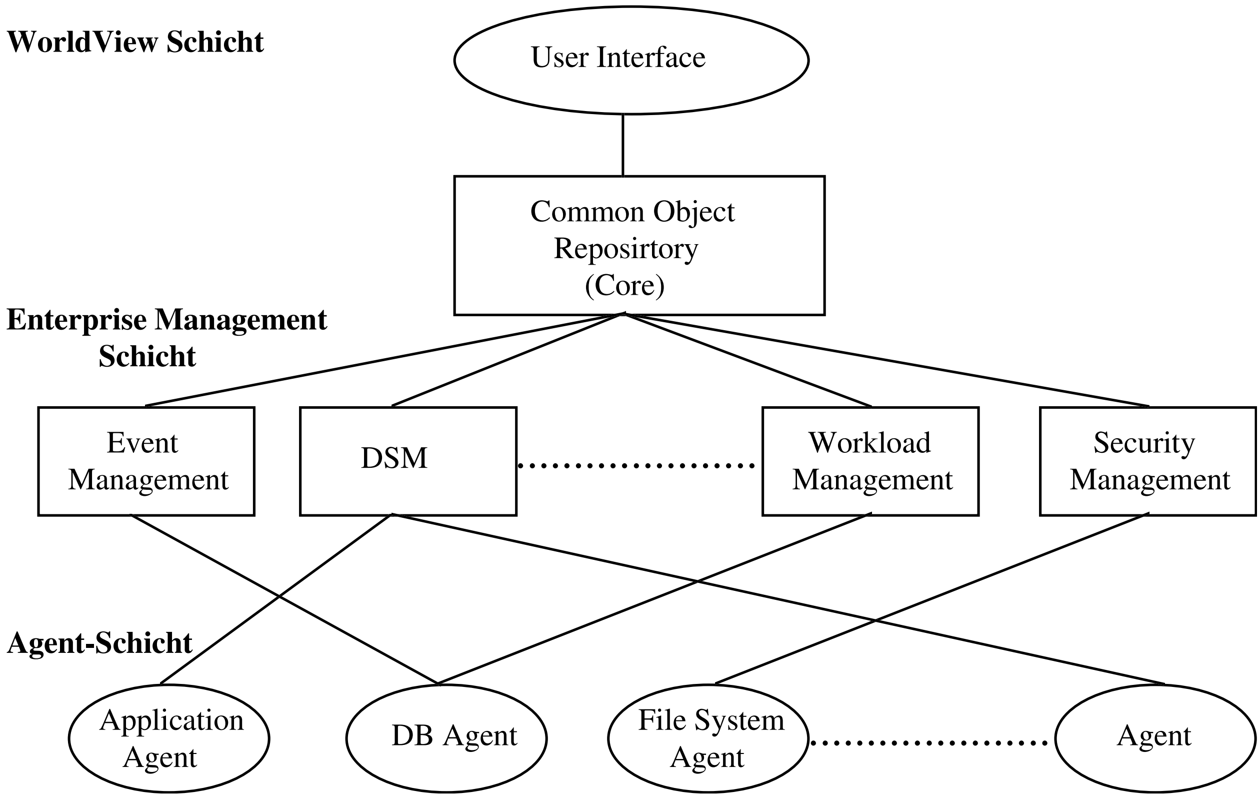


ABBILDUNG 2.1
DIE ARCHITEKTUR VON UNICENTER TNG

Unicenter TNG ist, obwohl es auf verschiedenen Plattformen, zum Beispiel Solaris 2.5/2.5.1 und HP-UX 10.20, einsatzfähig ist, eng mit dem Betriebsystem Windows NT verknüpft. Für andere Plattformen muß man damit rechnen, daß einige- wenn auch kleine - Einschränkungen bezüglich der Funktionalität gemacht werden müssen. Als bevorzugte relationalen Datenbanken, die als Basis für das objektorientierte Repository dienen, werden Microsoft SQL und CA Ingres eingesetzt.

Im Laufe des Praktikums wurde Unicenter TNG auf einem Windows NT-Rechner mit Pentium-III Prozessor und 128 MB RAM getestet.

2.2 Die Funktionen für DV-Management

In diesem Unterkapitel werden zwei Funktionen für DV-Management - Event Management und DSM - dargestellt, mit denen ich mich während meiner Arbeit beschäftigt habe.

2.2.1 Eventmanagement

Mit der Eventmanagement-Funktion kann man:
- Regeln definieren, die wichtige Aktivitäten, wie Meldungen, Protokolleinträge und Netzwerkereignisse erkennen
- automatische Reaktionen darauf einleiten und bei Bedarf sofortige Aktionen auslösen

Die Eventmanagement-Konsole dient als Schnittstelle zur Eventmanagement-Funktion. Die Konsole stellt ein spezielles Fenster in der GUI dar, mit dem man auftretende Systemereignisse überwachen kann.

Das Eventmanagement bietet Schnittstellen zum Eventprotokoll von Windows NT. Ereignisse, Kommandos und Meldungen, die vom Eventmanagement überwacht werden, lassen sich:
- kategorisieren
- unterdrücken
- auf der Eventmanagement-Konsole gesondert hervorheben
- in einem separaten Fenster der Eventmanagement-Konsole anzeigen
- zu einer zentralen Anzeige zusammenfassen
- nach Server, Benutzer, System und Workstation filtern
- zur automatischen Ausführung von Programmen als Reaktion auf ein Ereignis benutzen

Je nach Anforderungen läßt sich die Eventmanagement-Funktion so konfigurieren, daß Eventmeldungen entweder auf einzelnen Servern verarbeitet oder an einem zentralen Server oder andere Eventmanagement-Server bzw. Konsolen umgeleitet wird. Dank des Eventmanagements wird es einfacher, zusammengehörige Meldungen netzwerkweit zu erfassen, um sie auf eine zentralen Konsole anzuzeigen.

Funktionsweise des Eventmanagements

Mit dem Eventmanagement kann man bestimmte Meldungen kennzeichnen, die für den DV-Betrieb wichtig sind, und eine oder mehrere Aktionen, die bei Unicenter TNG Policy genannt werden, definieren, die bei jedem Auftreten der Meldung automatisch ausgeführt werden.
Das läßt sich am Beispiel eines Druckers erklären. Der Drucker sei sehr alt und hat ständig die Papierstau. Die Aufgabe ist, eine Aktion zu definieren, die hilft das Problem zu beheben, bevor die Benutzer sich beschweren. Dabei muß folgendes beachtet werden:

1. Werden beim Auftreten eines Problems die entsprechende Meldungen generiert? Im oben beschriebenen Beispiel schickt der Drucker die entsprechende Meldung zum Server und schreibt sie auch in das Windows NT System Event Log.
2. Jetzt muß entschieden werden, wie auf die Meldung reagiert werden soll:
   a) Meldung in der Konsole anzeigen
   b) Objektstatus in Repository entsprechend ändern
   c) Befehle ausführen
3. Meldungen filtern. Im Druckerbeispiel wird jedes mal eine Meldung geschickt, wenn eine Datei schon gedruckt ist. Da die Meldung irrelevant ist, wird sie mit dem Befehl ‘DISCARD’ verdeckt.
4. Policy mit Hilfe von Unicenter GUI implementieren.

Beispiel
Auf die Meldung „Printer message via port LPT1 : Please clear paper jam“ muß entsprechend reagiert werden, d.h. die Meldung muß rot angezeigt und der Printerstatus in Repository muß auf „Critical“ geändert werden:

SENDKEEP RED

COMMAND setstat –C Printer –N Critical

Datensichten

In Unicenter sind die Tools für die Betrachtung und Handhabung von Daten zur IT-Infrastruktur in allgemeine Kategorien unterteilt:
- Browser und ObjectView für die Anzeige von Informationen des Common Object Repository.
- Browser und Viewer für Daten, die von Agenten bereitgestellt werden.

Object View
Unicenter stellt Browser für Klassen, Objekte und für die Topologie zur Verfügung
Der Class Browser stellt Unicenter-Klassen und deren Eigenschaften dar. Er bietet eine Übersicht über alle Klassen im Repository.
Der Object View listet die aus den Unicenter-Klassen abgeleiteten Objekte auf und liefert Informationen zu Klassen und Objekten.

Agent View
Unicenter bietet verschiedene Tools zum Anzeigen von Agenten und ihrer dazugehörigen Daten.
Der MIB Browser ermöglicht Zugriff auf die Management-Information-Base (MIB) eines Agenten. Der MIB Browser kann allerdings keine Daten sammeln, um sie zur Weiterverarbeitung an den Leitstand Unicenter TNG (visueller Alarm) und an das Event Management zu schicken.

Das heißt auch, daß keine Schwellwerte gesetzt und überwacht werden können. *Es werden nur die aktuellen Werte einer MIB angezeigt.* Das Setzen und die Überwachung von Schwellwerten ist nur in Verbindung mit einem speziellen Agent möglich, der zusätzlich besorgt werden muß. Alternativ kann man mit Hilfe des SDKs ein entsprechender Agent selbst implementieren (siehe Kapitel 5).

Das Node View zeigt die verwalteten Ressourcen in Form einer grafischen Baumstruktur an, die die hierarchischen Abhängigkeiten zwischen den einzelnen Ressourcen verdeutlicht (s. Abblidung 2.3). Erfahrungsgemäß könnte man sagen, daß das Node View sehr aussagekräftig ist, denn hier erhält man sofort einen Überblick über die auf dem Host installierten Agenten (in der Abbildung 2.3 ist es ein NT System Agent) und über die Art und den Zustand von verschiedenen Ressourcen, z.B. „Monitored Services“, „Monitored Processes“, „Memory“ etc., die überwacht werden.
Die zu überwachenden Ressourcen sind für die verschiedenen Unicenter-Agenten unterschiedlich. Die Ressourcen, die von einem NT System Agent überwacht werden, sind im Kapitel 3.2.2 beschrieben.
Die Ressourcenzustandsänderung wird hier optisch angezeigt, indem die Farbe der entsprechenden Kästchen von Grün auf Rot oder umgekehrt geändert wird.

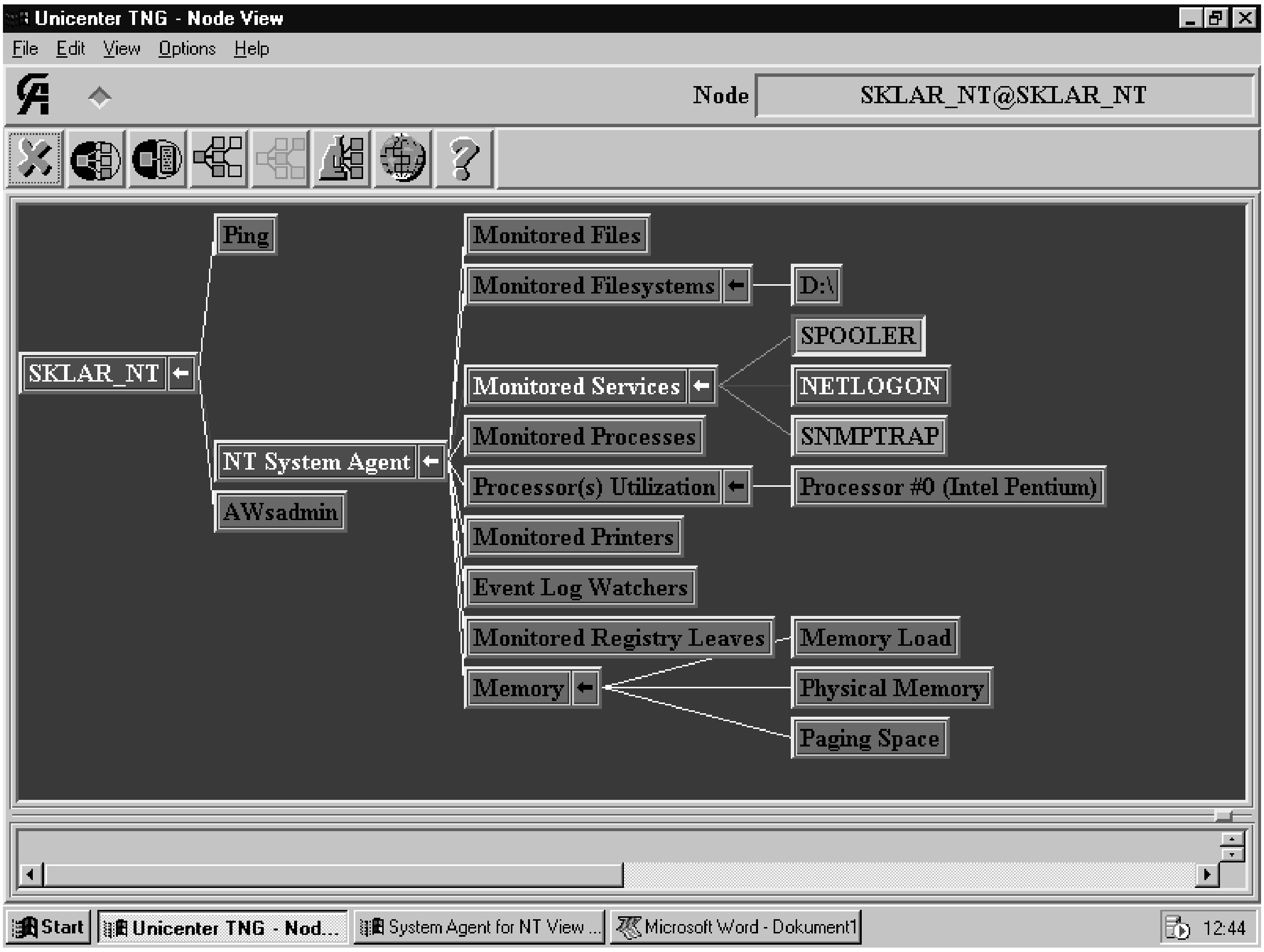


ABBILDUNG 2.2
NODE VIEW WINDOW

Mit Hilfe von **Agent Views** wird konfiguriert und angezeigt, welche Ressourcen wie z.B. Files, File Systems, Systemprozesse (d.h. NT-Diensten) u.s.w. , überwacht werden. Hier werden auch die Schwellwerte für die zu überwachenden Ressourcen eingestellt. Zum Beispiel zeigt die Abbildung 2.4 die momentan zu überwachenden Services : Spooler, Netlogon und SNMPTrap. In der Spalte „State“ sieht man die aktuelle Zustände von diesen Services und in der Spalte „Alert Level“ sieht man, auf was die obengenannten Ressourcen überprüft werden: *„Inaktive“* bedeutet, daß Ressourcenzustand Critical wird, wenn dieses Service läuft und wenn *„Aktive“* eingestellt ist, dann heißt es, daß das Service im Critical Zustand ist, wenn es nicht läuft.
Mit dem *„Add Service“* -Knopf können die neuen Ressourcen für die Überwachung hinzugefügt oder entfernt werden.
Mit dem *„Set Interval“*-Knopf werden die Poll-Intervalle eingestellt.

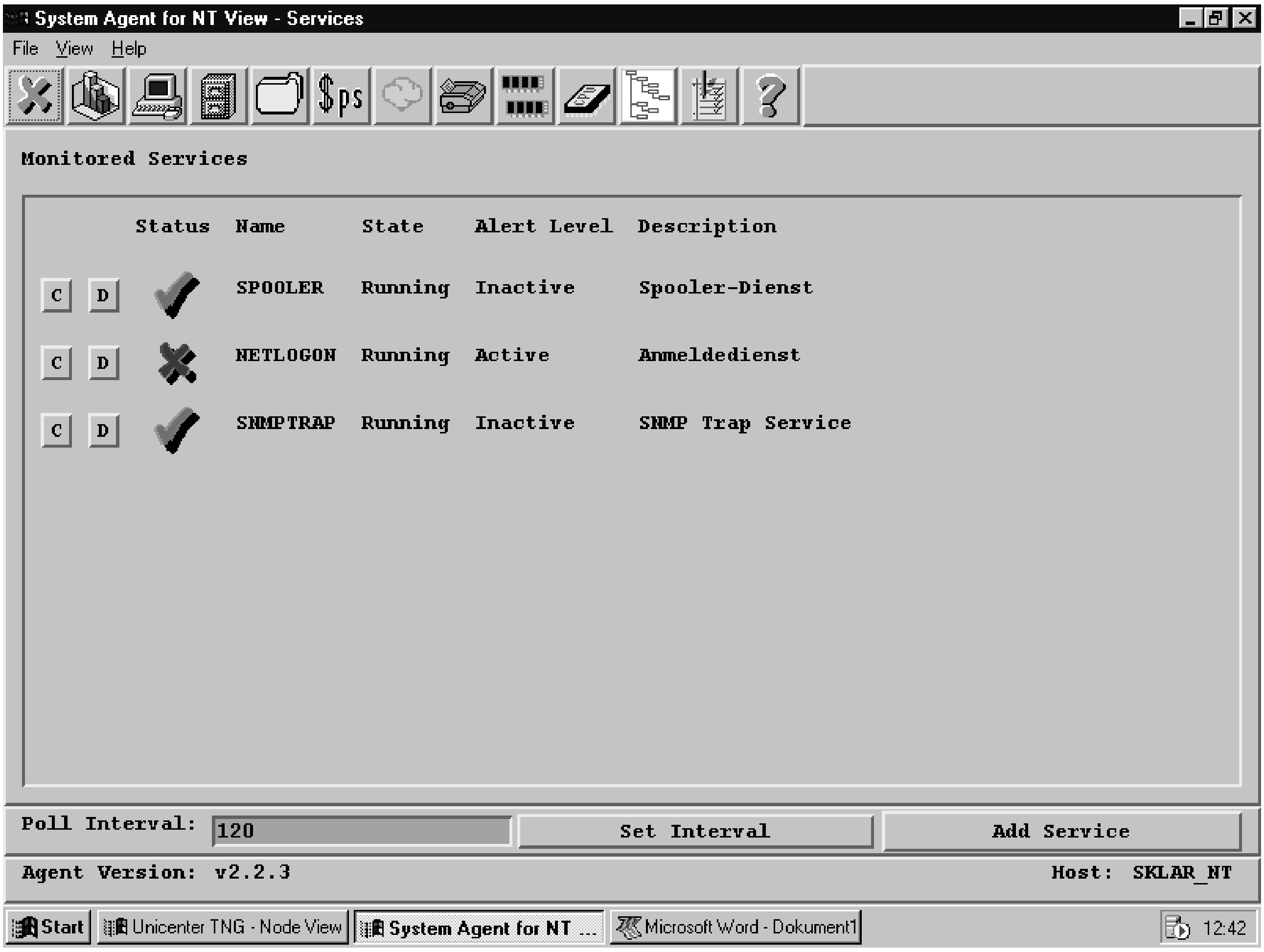


ABBILDUNG 2.3
SYSTEM AGENT VIEW - SERVICES

Die Abbildung 2.5 zeigt das „System Agent View - Summary" - der gesamte Überblick über alle überwachenden Ressourcen und ihre Zustände.

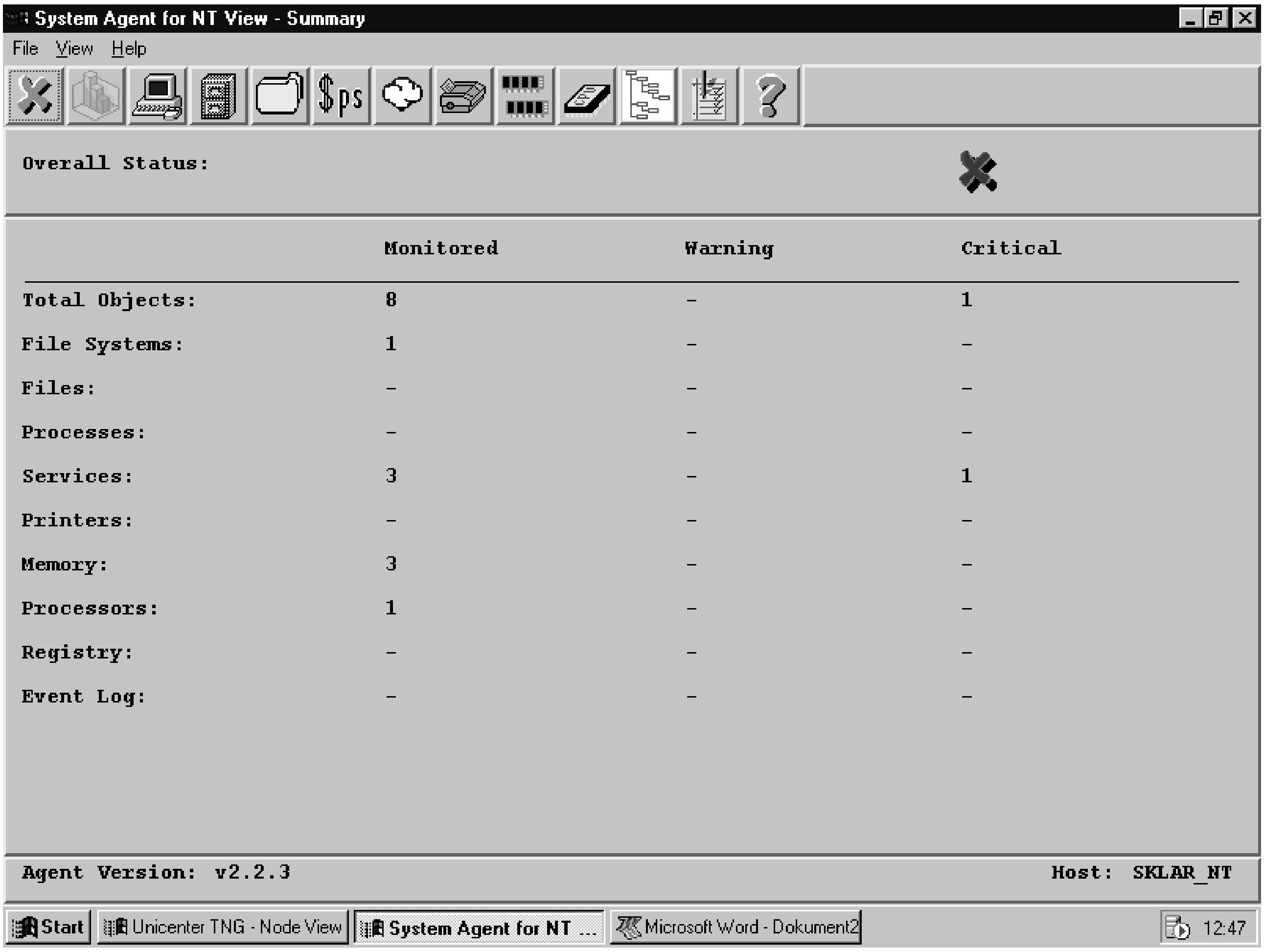


ABBILDUNG 2.4
SYSTEM AGENT VIEW - SUMMARY

Der Event Browser liefert detaillierte Informationen über Ereignisse, die Auswirkungen auf den von einem Agenten berichteten Status eines verwalteten Objekts hatten.
Mit dem Eventmanagement kann man Ereignisse kennzeichnen, auf die man reagieren will, und eine oder mehrere Aktionen festlegen, die automatisch angestoßen werden sollen. Wenn eine Meldung und ihre dazugehörige Aktionen einmal definiert sind, dann wird diese Aktion automatisch jedesmal ausgeführt, wenn Unicenter TNG das Ereignis feststellt.

2.2.2 Distributed State Machine (DSM)

DSM ist eine Schnittstelle zwischen dem Benutzer und den Agenten. Distributed State Machine untersucht den Status von Objekten im Netzwerk, indem es Agenten befragt.
Nehmen wir als Beispiel wieder ein Drucker. Er könnte On-Line oder Off-Line sein, er könnte zu viel Jobs oder kein Papier haben u.s.w. DSM spricht seinen Agenten an und stellt die Information in solcher Form dar, daß die Benutzer sie wahrnehmen können.

Man unterscheidet zwei Überwachungsmodelle:

- das synchrone Modell - das Polling Modell
- das asynchrone Modell - das Notification Modell

Beim **Polling Modell** werden Agenten in festgelegten Zeitintervallen abgefragt.

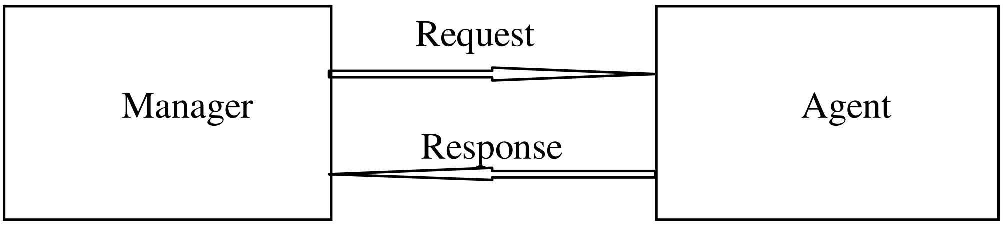


ABBILDUNG 2.5
DAS POLLING MODELL

Beim **Notification Modell** werden Manager von Agenten mittels Traps unaufgefordert verständigt, wenn etwas außergewöhnliches passiert.

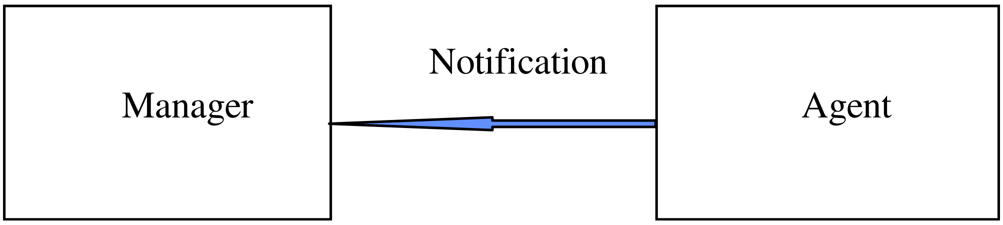


ABBILDUNG 2.6
DAS NOTIFICATION MODELL

Das Polling und das Notification Modelle ergänzen sich.

Es wäre zum Beispiel nicht effizient, einen Router-Agent alle 5 Sekunden abzufragen, ob das Error-Bit-Rate einen bestimmten Grenzwert überschritten hat. Auf der anderen Seite braucht ein Systemadministrator nach dem Ankommen eines Traps, daß das Error-Bit-Rate einen Grenzwert überschritten hat, mehr Information, um festzustellen, wo genau das Problem liegt. Dazu muß der Router-Agent abgefragt werden.

Das Polling wird auch aus dem Grund eingesetzt, daß der Systemadministrator nicht klären kann, was mit dem System geschieht, wenn keine Traps geschickt werden: entweder kann alles in Ordnung sein oder der Agent könnte schon längst ‘tot’ sein.

DSM-Policies sind Regeln, die bestimmen, welche Objekte beobachtet werden, wann und wie oft die Agenten abgefragt werden, wie Objekte zu finden u.s.w. Policies hängen von Systemkenntnisse ab.

Sollte die Abfragefrequenz hoch sein, werden die Fehler schneller gefunden aber Netzwerktraffic wird auch hohe. Unicenter TNG enthält einen Policy-Wizard, der dabei hilft, neue Policies zu erzeugen.

Um DSM zu starten, müssen folgende drei Schritte gemacht werden :
- DSM-Software installieren
- Teilnetze identifizieren, die beobachtet werden müssen
- DSM einschalten

DSM wird als Teil von Agent Technology installiert. Die Teilnetze, die zu überwachen sind, werden in der *Datei $AGENTWORKS_DIR\services\config\aws_wvgate\gwipflt.dat* beschrieben.
Zum Beispiel ein Eintrag „141.202.15.*“ sagt dem DSM, daß alle Hosts vom 141.202.15.0 bis 141.202.15.255 zu überwachen sind. Es ist „*.*.*.*.“ vorangestellt, das heißt, daß das ganze System zu überwachen ist.

2.2.3 Überblick über die anderen Funktionen für DV-Management

In diesem Unterkapitel werden die Funktionen für DV-Management, mit denen ich mich während meiner Arbeit nicht beschäftigt habe, grob beschrieben.

Workloadmanagement

Das Workloadmanagement beantwortet folgende Fragen:

- Woraus besteht die Systemlast (Workload)?
- Was sind die Spitzenbetriebszeiten?
- Wie lange dauert ein Prozeß (Job)?
- Welche Jobs laufen in der Regel wann?
- Gibt es Systemengpässe?
- Besteht die Möglichkeit zu einer besseren Verarbeitung der Arbeitslast?

Die Workloadmanagement-Funktion ist eine Lösung für die Automatisierung der Arbeitslastplanung. Sie steuert wichtige Abläufe, wie die Planung von Jobs, die Überwachung der Jobfolge, die Überwachung von Jobabbrüchen und die Einhaltung von Zeitvorgaben.

Das Workloadmanagement ist ein regelgestütztes System. Es verarbeitet automatisch die in der GUI definierten Regeln, welche Jobs wann, wo und wie verarbeitet werden sollen. Jobs und Jobsets werden vom Workloadmanagement automatisch am richtigen Tag, zur richtigen Zeit und in der richtigen Reihenfolge ausgewählt, geplant und abgesetzt. Basis dafür sind die in der Workloadmanagement-Datenbank gespeicherten Regeln.

Automatisches Speichermanagement (ASM)

Das automatische Speichermanagement umfaßt ein Dateisicherungs- und –archivierungssystem, das die Verlagerung von Dateien zwischen On-Line- und Backup-Datenträgern verfolgt.

Das ASM verfolgt die Sicherungs- und Archivirungsaktivität in einer gemeinsamen Dateidatenbank, mit der man Sicherungsdateien im Sicherungs- und Archivirungssystem auffinden kann.

Das ASM enthält Bandlabelling-Funktion, die automatischen Schutz und automatische Verfolgung von auf Band gespeicherten Sicherungs- und Archivierungsdateien bieten. Dank dieser Funktionalität ist das ASM in der Lage, Sicherungsbänder gegen vorzeitiges Überschreiben zu schützen, was zu einem Datenverlust führen könnte.

Das ASM handhabt den herkömmlichen Vorgang der Dateisicherung und –wiederherstellung mit Hilfe eines integrierten Systems, das Standort, Status und Historie jeder Sicherung verfolgt. Das ASM verfügt außerdem über spezielle Steuerelemente, die dafür sorgen , daß der Sicherungsdatenträger nicht versehentlich oder absichtlich überschrieben wird.

Sicherheitsmanagement

Das Sicherheitsmanagement ist eine regelgestützte Einrichtung zum Schutz der IT-Infrastruktur:

- unberechtigte Personen davon abhalten, Zugang zum System zu erhalten,
- sicherstellen, daß sensible Systemsteuerdaten und –ressourcen nur von berechtigten Personen genutzt werden,
- Daten und Ressourcen gegen Rechtemißbrauch durch Systemanwender schützen,
- und die Daten und Ressourcennutzung überprüfen und revidieren.

Problemmanagement

Das Problemmanagement bildet einen Rahmen für die Verwaltung der folgenden Probleme:

- Fehlererkennung
- Fehlerlokalisierung.
- Fehlerlösung

Innerhalb das Problemmanagement gibt es drei große Bereiche:

- Komponentendefinition
- Problemdefinition
- Machine-Generated Problem Tracking (MGPT)

Komponentendefinition dienen zur Definition der Konfiguration des Systems, einschließlich Hardware, Software, und sonstige Ausstattungen wie Haustechnik (Klimaanlage), Telekommunikationsanlagen, Sicherheitssysteme und andere Komponenten, die man innerhalb eines Unternehmens verfolgen will. Zusammen mit der Problemdefinition stellen Komponentendefinitionen Informationen zur Verfügung, anhand derer analysiert werden kann, welche Auswirkungen der Ausfall einer Komponente auf eine andere hat.

Problemdefinition werden in das Problemmanagement von den Mitarbeitern des Help Desk manuell und von der Machine-Generated Problem Tracking (MGPT) automatisch eingetragen. Ein Problem stellt jeder gemeldete Zwischenfall dar, der Nachforschung und Einleitung von Maßnahmen erfordert.

Machine-Generated Problem Tracking (MGPT) ermöglicht die automatische Erstellung von Problemeinträgen auf, Basis der vom Eventmanagement überwachten Aktivitäten. MGPT-Regeln dienen zur Erkennung von Problemen auf einzelnen Hosts, innerhalb von Anwendungen und im Netz.

Kapitel 3 Agent Technology: Architektur und Funktionalität

In diesem Kapitel wird es auf die Architektur von Agent Technology sowie auf den Aufbau ihrer einzelnen Schichten eingegangen und die Aufgaben und Features von Unicenter System Agent geklärt. Anschließend werden Neugent dargestellt und der Unterschied zwischen Neugent und den herkömmlichen Agenten geklärt.

3.1 Architektur

Agent Technology ist eine Unicenter-Komponente, die zusätzlich auf dem Manager-PC installiert werden soll, um die Kommunikation mit Agenten zu ermöglichen. Agenten sind Software, die an Ressourcen, z.B. Speicher, Dateien, Anwendungen etc., gebunden werden können, um die Information von ihnen zu sammeln und sie zur Verarbeitung weiterzuleiten.

Die Architektur von Agent Technology umfaßt (siehe Abbildung 3.1) **:**
- Agent Schicht
- DSM-Schicht
- WorldView Schicht

Die Aufgabe der Agent-Schicht ist es, die Informationen über die Ressourcen, an die die entsprechenden Agenten gebunden sind, zu ermitteln, sie zu Traps zu codieren und der DSM-Schicht zur Weiterverarbeitung zur Verfügung zu stellen.

Die Aufgabe der DSM Schicht ist es, die SNMP Traps von der darunterliegenden Agent-Schicht entgegenzunehmen, sie zu entschlüsseln und in die Object Common Repository der WorldView Schicht zu schreiben.

Die Aufgabe der WorldView Schicht ist es, die Eigenschaften von zu verwaltenden Objekten in Object Common Repository zu halten und die Objekte selbe graphisch auf der Oberfläche darzustellen.

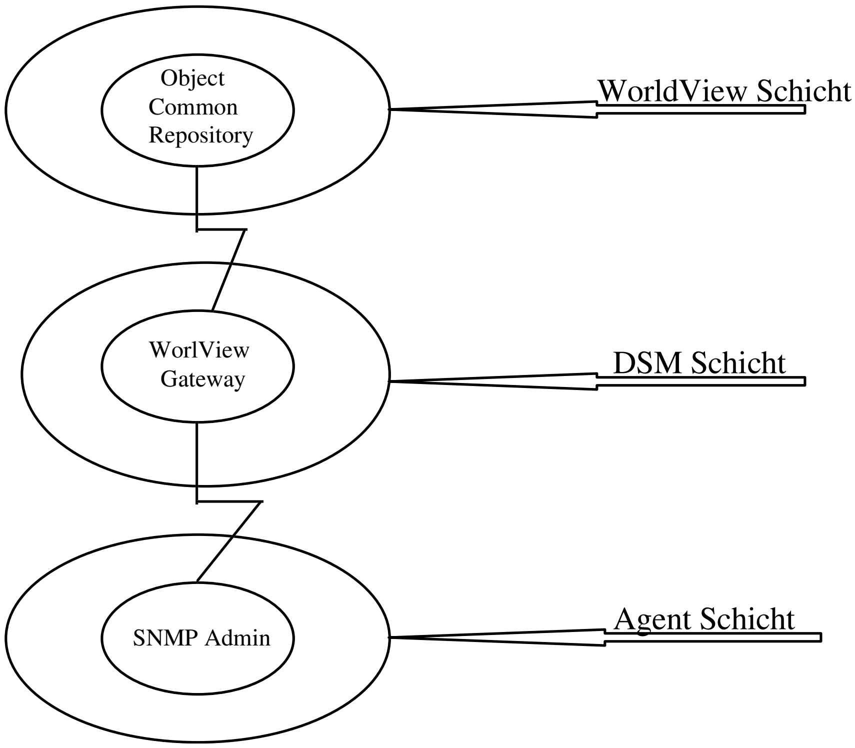


ABBILDUNG 3.1
AGENT TECHNOLOGY ARCHITEKTUR

Die **Agent Schicht** (s. Abbildung 3.2) befindet sich auf den zu managen Hosts und umfaßt

- SNMP Admin
- SADMIN Store
- Distributed Service Bus (DSB)
- Agent

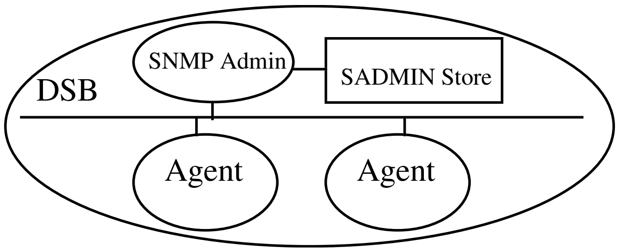


ABBILDUNG 3.2
AGENT SCHICHT

Der **Distributed Service Bus** (DSB) ist ein Bus innerhalb der Unicenter TNG Agentenarchitektur, der zur Informationsaustausch dient.

Die **Agenten** (z.B. NT System Agent, OS/2 Agent oder Ingres Agent) werden lokal auf den zu überwachenden System installiert, an den DSB angedockt und übergeben ihre Meßwerte an den **SNMP ADMIN**. Dieser chiffriert diese Information zu Traps und versendet sie mittels SNMP an die zuständige Distributed State Machine (s. unten).
SADMIN Store ist der Speicher für die MIB-Informationen.

Die DSM Schicht (s. Abbildung 3.3) beinhaltet

- Distributed State Machine (DSM)

- SNMP Gateway
- DSB
- Object Store
- Domain Manager
- WorldView Gateway

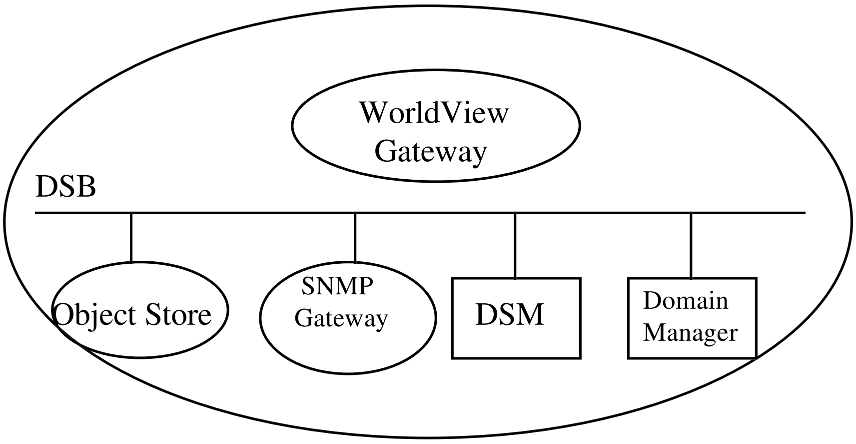


ABBILDUNG 3.3
DSM SCHICHT

Die **DSM** überprüft die Verfügbarkeit der installierten Agenten und nimmt deren SNMP Traps entgegen. Die DSM läuft meistens auf dem Unicenter TNG Server. Wenn man mehrere in der Fläche verteilte Subnetze einsetzt, so kann man in den einzelnen Lokationen separate DSM installieren.

SNMP Gateway funktioniert wie eine Poststation - hier werden SNMP Traps entgegengenommen, entschlüsselt und via Distributed Service Bus (DSB) zur Weiterverarbeitung weitergereicht.

Object Store funktioniert als Definitionsspeicher für die Klassenparameter.

Domain Manager funktioniert als Entscheider, der auf Basis eines Definitionsspeichers für die Bewertungsregeln, ob der Trap einen Normalstatus, Warningstatus oder einen anderen zustand bedeutet.

WorldView Gateway schreibt die Objectzustände in die Object Common Repository der WorldView Schicht.

Die WorldView Schicht (s. Abbildung 3.4) umfaßt
- WorldView Interface
- Object Common Repository

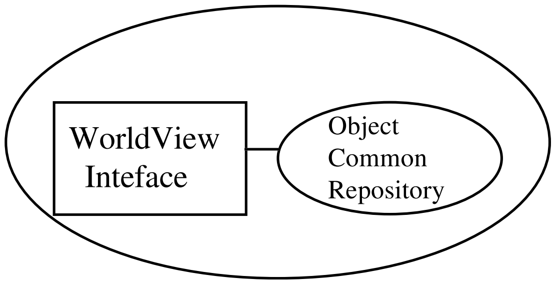


ABBILDUNG 3.4
WORLDVIEW SCHICHT

WorldView Schicht stellt PC's, Datenbanken, Applikationen und auch andere Ressourcen durch die WorldView 2-D und 3-D maps dar. Diese Objekten werden von dem Object Common Repository gewonnen.

Object Common Repository ist eine Aufbewahrungstelle für alle in der Umgebung erkannten Objekte und ihre Eigenschaften.

3.2 Agent Funktionalität

Die **Agenten,** die lokal auf den zu überwachenden System installiert sind, sind Software, die Informationen über die IT-Infrastruktur durch Fernzugriff-Überwachung und -Steuerung von Ressourcen, sammeln. Ein wesentliches Merkmal der Agenten ist die Möglichkeit, sie an Ressourcen zu binden, um so spezifische Informationen über die jeweilige Ressource zu sammeln und die Ressource zu verwalten.

Die Aufgabe von Agenten besteht darin, die wichtigen Daten über die Ressourcen, an die sie gebunden werden, für die Management-Komponente zur Verfügung zu stellen und die Problemursachen sicherzustellen, bevor sie Betriebsabläufe negativ beeinflussen.

Features von NT System Agent

Unicenter stellt eine Vielzahl von Agenten zur Verfügung: Windows95 Agent, Oracle Agent, SQL Server Agent, OS/2 Agent usw. Die Ressourcen, die von Unicenter Agenten überwacht werden, unterscheiden sich entsprechend. Während meines Praktikums habe ich mich mit dem **NT System Agent** auseinandergesetzt.

NT System Agent überwacht und managt folgende Ressourcen:

- **File Systems**. Das File System wird auf vorhandenen freien Platz überprüft und das Ergebnis wird mit benutzerdefinierten Daten und kritischen Tresholds verglichen. Man kann auch das File System Format (FAT oder NTFS) und die Partitionsgröße ermitteln.
- **Files**. Fileattribute, wie Filegröße und Time-Stamp Change werden überwacht. Man kann für sie Schwellwerte definieren.
- **Systemprozesse**. Die Daten von allen Systemprozessen (NT-Diensten) werden von Agent in Echtzeit gesammelt. Das läßt feststellen, ob wichtige Prozessen plötzlich angehalten worden sind, das heißt man kann hier nur *„Alert Level"* überwachen (s. Abbildung 3.4). Man ist hier nicht in der Lage, Systemprozesse auf andere kritische- und benutzerdefinierte Daten zu überprüfen.
- **Printers**. Printers, die direkt an Windows NT angeschlossen sind, werden auf Jobsanzahl kontrolliert, ob diese ein kritisches Niveau überschritten hat. Man ist hier nicht in der Lage, einen Netz-Printer zu überwachen - dazu werden spezielle Agenten, z.B. ein HP-Agent, benötigt.

- **Memory**. Hier werden die Größe der physischen Speicher sowie die des Swap-Bereichs überwacht.
- **Event Log**. Windows Event Log wird auf die *Existenz* von Security-, System- und Anwendungsereignisse kontrolliert. Das heißt, man definiert die Quelle des Ereignisses, z.B. SNMP oder MSSQLServer, und bekommt die Eventsanzahl angezeigt.
- **Prozessor.** Hier kann nur *„CPU Utilization"* überwacht werden.

Unterstützte Betriebsysteme:

- Windows 95
- Windows 98
- Windows NT (Intel und Alpha)
- NetWare
- OS/2, OS/400, OS/390
- Tandem NSK
- Open VMS (VAX und Alpha)
- UNIX (IBM, Digital, Data General, Sequent, Sun [Intel und SPARC], NCR, Hewlett Packard, ICL, SLI, SGI)

Systemanforderungen:

- OS/390 1.3 oder höhe
- TCP/IP 3.2 oder höhe
- Unicenter TNG 2.1

Neugent

Der Unterschied zwischen **Neugent** und den herkömmlichen Agenten liegt darin, daß noch nicht aufgetretene Probleme von Neugent entdeckt werden, bevor sie einen negativen Einfluß auf Betriebsabläufe ausüben können; das heißt, daß die zukünftigen Probleme prognostiziert werden können.

Neugent können voraussagen, wie die Änderung von Systemparametern das System beeinflussen können. Obwohl diese Parameter nicht immer das System beeinflussen, kann nicht rechtzeitige Problembehebung zum Systemschaden führen.

Neugent können das System »lernen«, indem sie historische Systemdaten beobachten und analysieren. Aus Tausenden Parametern wird ein persönliches Systemprofil erstellt, das ein normalfunktionierendes System repräsentieren kann.

.

Kapitel 4 Die betriebsablauforientierte Sicht

In diesem Kapitel werden die Funktionsweise der betriebsablauforientierten Sicht sowie ihre Vor- und Nachteile beschrieben. Anhand eines Beispielsbetriebsablaufs werden die einzelnen Schritten, die von Modellierung zur Umsetzung eines Beispielsbetriebsablaufs in Unicenter TNG führen, dargestellt.

Ein Nebeneffekt des informationstechnologischen Fortschritts sind die gewaltigen Datenmengen, vor denen Unternehmen überflutet werden. Zweck der Informationstechnologie ist es, die zur Erfüllung einer Aufgabe benötigten Daten bei Bedarf verfügbar zu machen. Dies erfordert oftmals die Pflege von großen Datenmengen. Alle Daten sind in der Regel vorhanden, sie jedoch zu finden, strukturieren und einzusehen ist ein mühsamer, fehleranfälliger, zeitaufwendiger Prozeß [wwwCA].

Mit dem sogenannten **Bussines Process View**, der betriebsablauforientierten Sicht, bietet Unicenter TNG hierfür eine Lösung.

Die betriebsablauforientierte Sicht umfaßt alle Komponenten, die für einen Betriebsablauf wichtig sind, die man überwachen möchte. Dank der betriebsablauforientierten Sichtweise lassen sich Unternehmensabläufe mit dem entsprechenden Ressourcenbedarf verbinden und die betroffenen Ressource innerhalb eines bestimmten betrieblichen Funktionsbereichs überwachen. Sollte sich eine der Komponenten in einem kritischen Zustand befinden, wird es sofort sichtlich gemacht, indem das entsprechende Symbol in einer entsprechenden Farbe angezeigt wird, ohne daß man weitere Komponenten berücksichtigt werden muß.

Ein Beispiel: Ein IT-Manager wird darüber informiert, daß die Aufarbeitung der Lohn- und Gehaltsabrechnung erheblich in Verzug ist. Ohne durchgängiges DV-Management wird er nicht sofort wissen, ob das Problem bei der Lohn- und Gehaltsabrechnungs-Anwendung selbst, dem Netzwerk, einer Datenbank, dem Host, auf dem die Anwendung läuft, oder woanders zu suchen ist - manchmal wird er noch **nicht einmal alle Faktoren**, von denen der Betriebsablauf Lohn- und Gehaltsabrechnung abhängt, genau kennen.

Die betriebsablauforientierten Sichten sind vollständig benutzerdefinierbar und können alles darstellen (oder enthalten) was man für geeignet oder relevant hält.
Daß könnte beispielsweise Debitorenbuchhaltung sein, die Softwareanwendungen z. B. (auf zwei Unix-Servern), 50 Workstations, zwei Datenbanken und zahlreiche andere Prozesse umfaßt. Beim Einsatz von System Management-Tools kann man über eine grafische Benutzeroberfläche (GUI) den Status der Server, mit einer anderen den Status der Datenbanken verfolgen, usw. Man muß die verschiedenen Informationen, die man von den unterschiedlichen GUIs erhält, zusammenfügen, und den Status des Debitorenbuchhaltungsablaufs festzustellen - und man muß diese Schritte jedesmal wenn man den Status wissen möchte, wiederholen. Mit Business Process View (BPV) lassen sich diese unterschiedlichen Daten zum Status von Software, Hardware und Prozeß logisch in einer Sicht zusammenfassen.

Das Konzept der betriebsablauforientierten Sichten in Unicenter TNG beruht auf der Kombination zweier Perspektiven oder Sichten (siehe Abb. 5.1) [RiSt1998]:
- Horizontale Sicht
- Vertikale Sicht

Horizontale Sichten (s. Abbildung 5.1) können Strukturen und Prioritäten aus DV-Perspektive wiedergeben. Die nachfolgende Abbildung zeigt die horizontale Sicht. Dazu gehören beispielsweise getrennte Gruppen für Anwendungen, Datenbankverwaltung, Systemprogrammierung und Netzwerkmanagement.

Vertikale Sichten (s. Abbildung 4.1) geben Strukturen und Prioritäten aus betriebswirtschaftlicher Perspektive wieder.

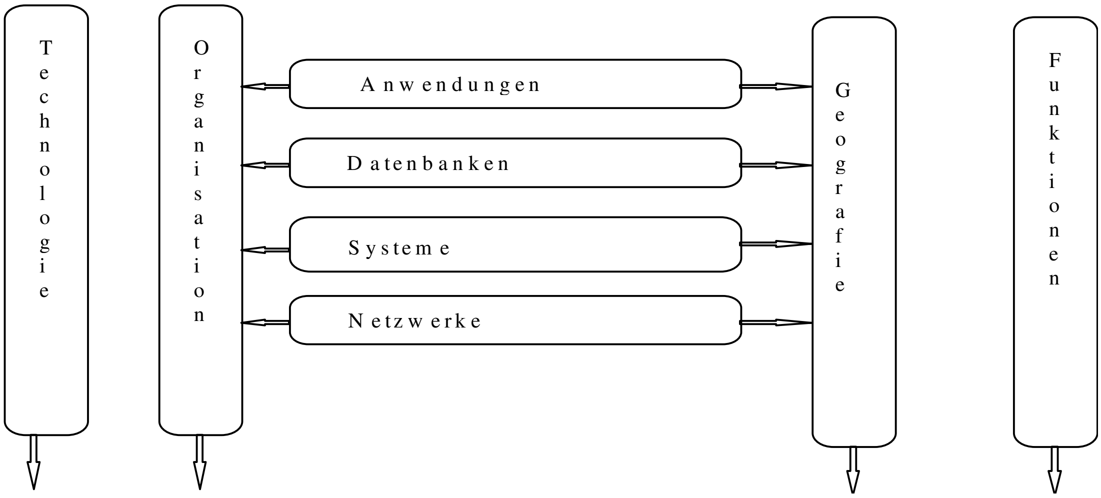


ABBILDUNG 4.1
ARCHITEKTUR VON BETRIEBSABLAUFORIENTIERTEN SICHTEN [RIST1998]

Die betriebsablauforientierten Sichten ermöglichen eine vertikale Sicht über horizontale Sichten. Betriebsablauforientierte Sichten werden mit Hilfe der WorldView-Schicht erstellt. Diese Sichten enthalten in der Regel die IT-Ressourcen, die für den Benutzer bei der Ausführung seiner täglichen Aufgaben wichtig sind . Einmal definierte Sichten können bei Bedarf geändert werden, wobei IT-Ressourcen hinzugefügt oder herausgenommen werden können.

Die Praktikumsaufgabe, einen **Printer-Betriebsablauf,** der aus einem HP-Printer „*gutenberg*“ und einem WindowsNT-Host „*pcheger5*“ besteht, zu überwachen, wird durch Modellieren und Erstellung der dazugehörigen betriebsablauforientierten Sichtes gelöst.

Zuerst überlegt man sich, **welche Betriebsablaufkomponenten** (weiter Komponenten oder Ressourcen) **ein Printer-Betriebsablauf enthalten kann**. Ein wirklicher Printer-Betriebsablauf kann mehrere Komponenten enthalten, z.B. ob es ausreichend Papier in allen Fächern gibt, ob es schwarz-weiß- oder farb gedrückt werden kann, ob Printer On-Line oder Off-Line ist, welcher User im Moment darauf drückt, ob der Spooler-Dienst auf dem Host läuft, ob es Papierstau gibt usw.

Dann macht man sich die Gedanken, **welche Komponenten man überwachen möchte** und **wie ihre Zustände** (Normal, Warning oder Critical) **auf den Gesamtzustand der betriebsablauforientierten Sicht auswirken können**.

Alle in meinem Beispiel überwachten Ressourcen kann man in zwei Gruppen aufteilen: die, die sich auf das Host „pcheger5“ und die, die sich auf den Drucker „gutenberg“ beziehen. Die Überwachung der **Ressourcen der ersten Gruppe** soll sicherstellen, daß WindowsNT-Dienste - z.B. Spooler-Dienst oder Arbeitsstationdienst - die für das Drucken zuständig sind, laufen. Die Überwachung **der zweiten Ressourcengruppe** sorgt dafür, daß die Problemen, die von der Drucker’s Seite ausgehen können, wie z.B. ob der Drucker On-Line ist oder ob es kein Papierstau vorliegt, rechtzeitig

entdeckt werden. Es sollte aber klar sein, daß es hier nur um ein Beispiel handelt: ein echter Betriebsablauf ist viel komplexer.
In meinem Beispiel überwache ich die folgenden Komponenten (s. Abbildung 4.2):

- „ping on Host" - ob „pcheger5" „am Leben ist" .
  Falls die Komponente ausfallt, soll der Gesamtzustand Critical werden.
- „ping on Printer" - ob „gutenberg" „am Leben ist" .
  Beim Ausfall dieser Komponente soll der Gesamtzustand ebenso Critical werden.
- „Agentservice on Host" - ob System Agent auf „pcheger5" funktioniert.
  Sollte die Komponente ausfallen, soll der Gesamtzustand Warning werden, denn es ist immer noch möglich zu drucken.
- „Spooler on Host" - ob der Spooler-Dienst auf „pcheger5" läuft.
  Beim Ausfall dieser Komponente soll der Gesamtzustand Critical werden.
- „Arbeitsstationdienst" - ob die Printerfreigabe funktioniert.
  Der Gesamtzustand soll Warning werden, wenn der Arbeitsstationdienst auf dem Host nicht mehr läuft.
- „Paper in Box 1" - ob „gutenberg" ausreichend Papier in Box 1 hat.
  Es sollte eine logische „Oder"-Beziehung zwischen dieser und „Paper in Box 2"-Komponente definiert werden. Das heißt, nur dann, wenn die beiden Komponenten im Critical Zustand sind, soll der Gesamtzustand auch Critical sein, wenn aber nur eine von beiden Critical und eine andere Normal ist, dann soll der Gesamtzustand Warning werden, denn es kann immer noch gedruckt werden.
- „Paper in Box 2" - ob „gutenberg" ausreichend Papier in Box 2 hat.
- „Printer On-Line" - ob „gutenberg" On-Line ist.
  Sollte die Komponente Critical sein, ist der Gesamtzustand auch Critical.
- „Paper Jam" - ob es Papierstau gibt.
  Ist die Komponente Critical, dann soll der Gesamtzustand ebenso Critical werden.

Der nächste Schritt ist es, sich zu überlegen, welche **Anforderungen an der betriebsablauforientierten Sicht** sinnvoll zu definieren sind**:**

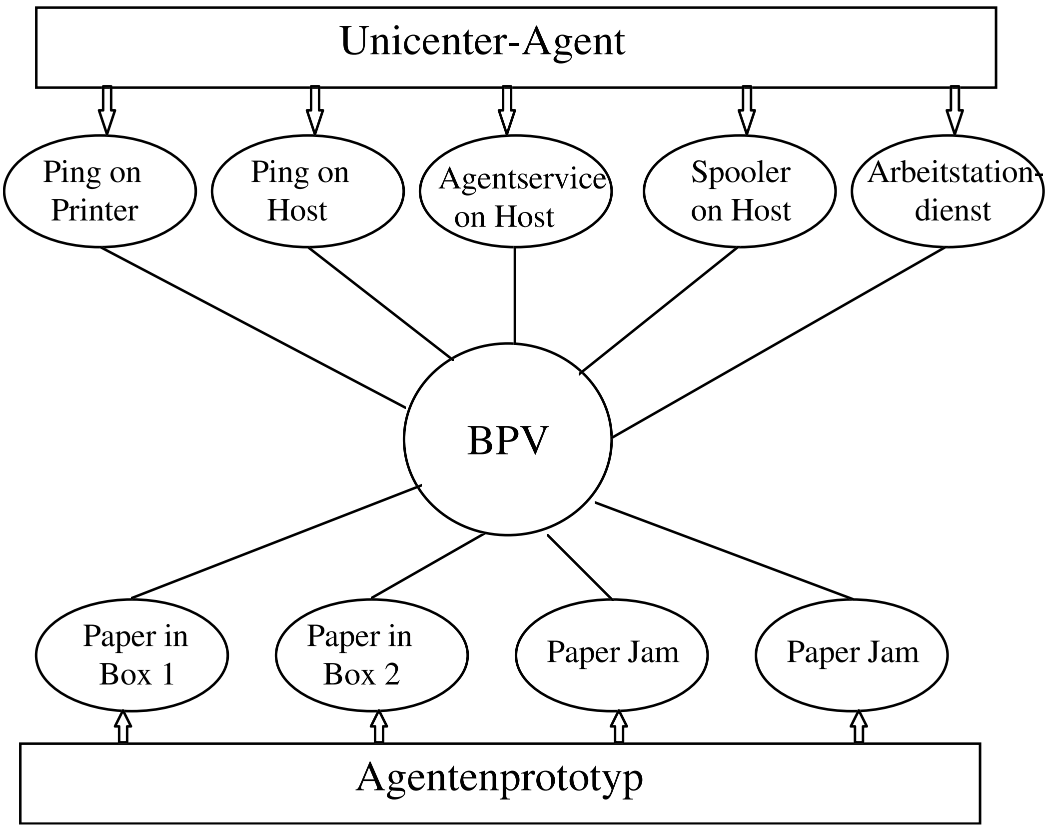
- Ermittlung des Gesamtzustandes der betriebsablauforientierten Sicht:
  Das heißt, man möchte wissen, ob man auf dem „gutenberg" überhaupt drucken kann.
- Identifikation fehlerhafter Komponenten:
  Sollte man nicht in der Lage sein zu drucken, möchte man wissen, die Zustände von welchen Komponenten dies hervorgerufen haben.
- Modellierung der Beziehungen zwischen Komponenten:
  - Und/Oder-Beziehungen
  - Redundanz
  - Lastverteilung

  Wenn der Gesamtzustand vom Zusammenwirken von mehreren Komponenten abhängig ist, möchte man die logische Beziehung zwischen Komponenten und dem Gesamtzustand definieren.

Nachdem die Anforderungen definiert sind, überlegt man sich, wie der oben beschriebene **Beispiel-Betriebsablauf in Unicenter umgesetzt werden kann,** welche Tools, welche Mittels für die Betriebsablauf-Umsetzung angeboten werden.

Die Tools, mit denen man die Komponenten, die sich auf Host „pcheger5" beziehen (s. Abbildung 4.2), überwachen kann, sind da. Das sind der **Unicenter System Agent** und **das Agent View**. Der Agent, der auf „pcheger5" installiert wird, sendet in der vordefinierten Zeitabständen die Information, wie sich die zu überwachenden Dienste verhalten.

Mit dem Agent View (s. Abbildung 2.4), das auf dem Manager PC installiert wird, kann man definieren, welche Dienste zu überwachen sind, das Pollintervall und in welchem Fall eine Error-Meldung generiert werden soll, das heißt, ob es signalisiert werden soll, ob der Dienst läuft oder nicht läuft.



ABBLIDUNG 4.2
BEISPIEL-BETRIEBSABLAUF

Um die vier Komponenten, die sich auf „gutenberg" beziehen, überwachen zu können, ob der „gutenberg" On-Line ist oder ob er genug Papier hat, soll entweder ein HP Agent installiert oder ein geeigneter MIB-Browser, mit dem man die Schwellwerte über die MIB-Variable setzen und überwachen kann, eingesetzt werden. Da es kein HP-Agent in der Standardlieferung ist und kein MIB-Browser vorhanden ist, der die Schwellwerte überwachen kann (s. Kapitel 3, Datensichten), gibt es eine andere Möglichkeit, einen eigenen Agent zu erstellen, der in der Lage ist, interessierende Werte zu überwachen (s. Kapitel 5).

Im nächsten Schritt wird es analysiert, ob die vorher **definierte Anforderungen an der betriebsablauforientierten Sicht erfüllt werden.**
Wie man in der Abbildung 4.3 sehen könnte, ist der Gesamtzustand des Betriebsablauf sofort zu ermitteln - das Symbol, das der Gesamtzustand des Betriebsablaufs repräsentiert, wird auf dem Betriebsablauf-Map in entsprechender Farbe (grün für Normal Zustand, rot für Critical Zustand und gelb für Warning Zustand) angezeigt. Ebenso leicht sind die fehlerhaften Komponenten festzustellen: nach dem Doppelklick auf das Betriebsablauf-Symbol wird der entsprechende Map mit den im Betriebsablauf enthaltenen Komponenten angezeigt. Sobald eine der zu überwachenden Komponenten im Critical oder Warning Zustand ist, wird eine Error-Meldung gesendet und das Komponentensymbol wird rot bzw. gelb (s. Abbildung 4.3).

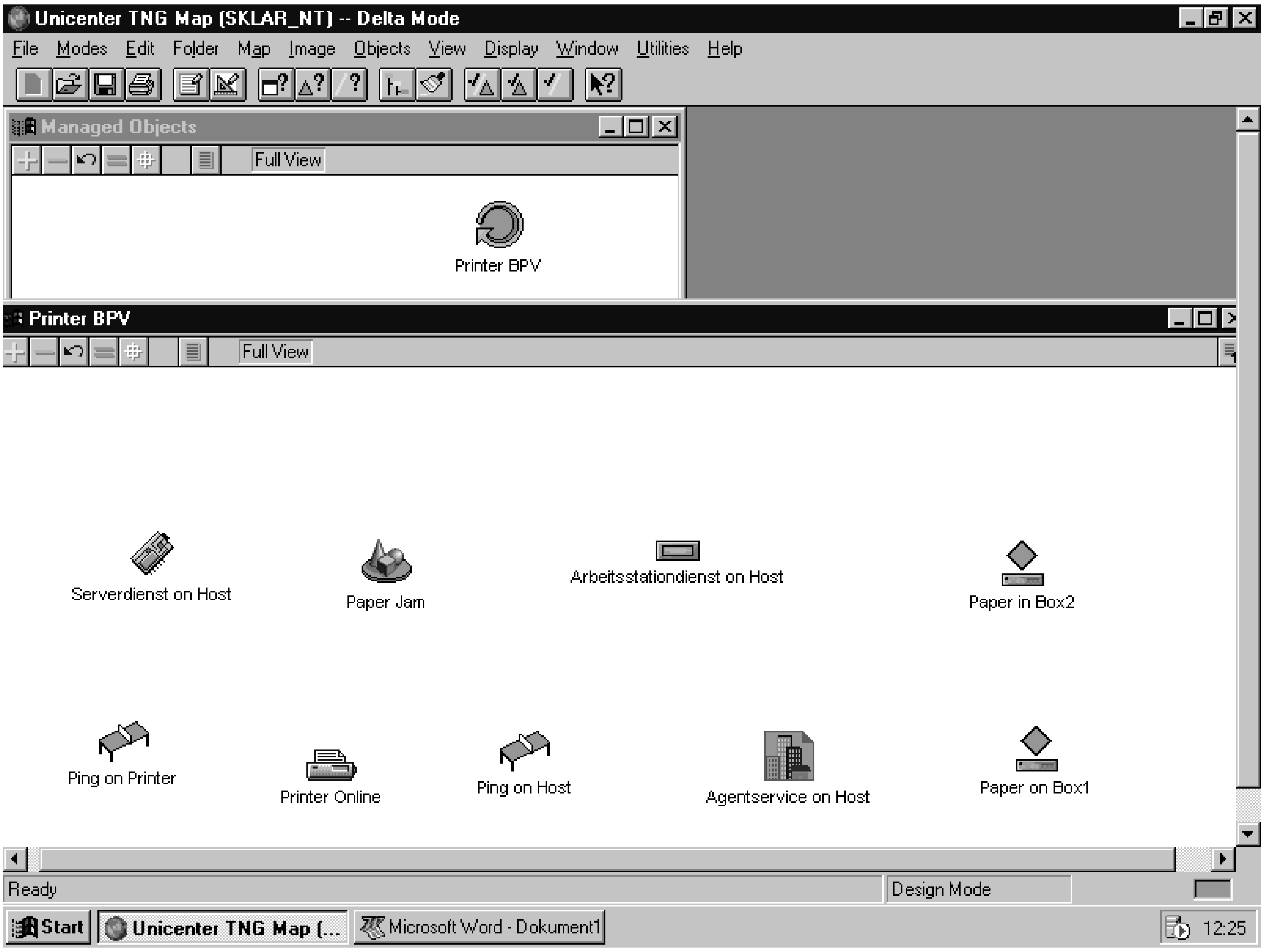


ABBILDUNG 4.3
PRINTER-BETRIEBSABLAUF

Leider sind die logischen Beziehungen in der betriebsablauforientierten Sicht nicht implementiert. Betriebsablauforientierte Sicht nimmt den „schlechtesten“ Zustand von allen Zuständen aller Komponenten auf, die in einer Sicht enthalten sind. D.h. wenn alle Komponenten im Normal Zustand sind und nur ein im Warning Zustand, dann geht die ganze Sicht in den Warning Zustand über.

Betrachten wir z.B. folgende betriebsablauforientierte Sicht BS1:

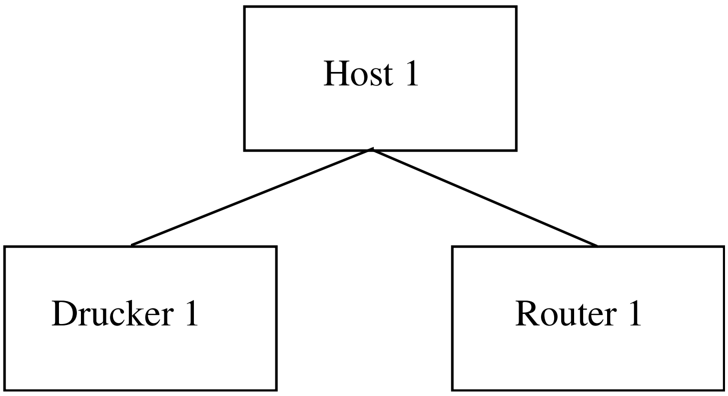


ABBILDUNG 4.4

Seien „Host" und „Drucker" im Normal Zustand und „Router" im Critical Zustand. Dann geht die ganze Sicht in den Critical Zustand über, also sie wird rot und man könnte denken, daß der Druckvorgang nicht mehr möglich ist, obwohl daß nicht der Fall ist. Eine mögliche Lösung wäre die Sicht so zu konfigurieren, daß sie in den Warning Zustand übergeht, also wird gelb, so daß man sieht, daß etwas nicht in Ordnung ist aber der Druckvorgang immer noch funktioniert. Das läßt sich aber leider nicht konfigurieren.

Das Beispiel geht weiter:
Sei BS2 eine weitere betriebsablauforientierte Sicht.

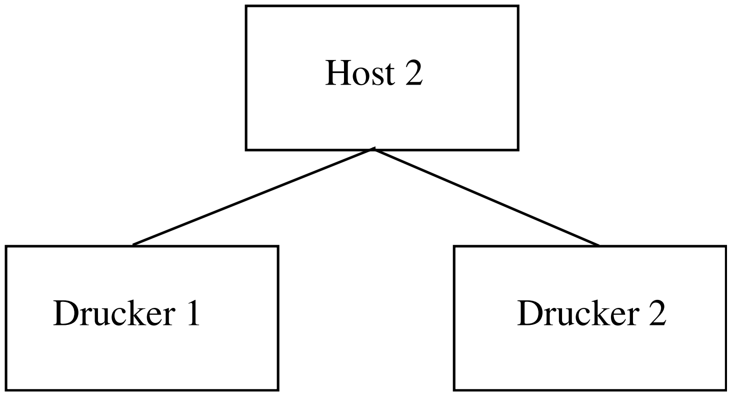


ABBILDUNG 4.5

Der Drucker 1 ist an zwei Druckvorgängen beteiligt: Host 1 druckt auf Drucker 1 und Host 2 druckt auch auf Drucker 1. Sollte Drucker 1 ausfallen, dann gehen die beiden Sichten BS1 und BS2 in den Critical Zustand über, obwohl der Druckvorgang in der BS2 nicht abgebrochen ist, denn Host 2 immer noch auf Drucker 2 drucken kann. D.h. BS2 sollte im Warning Zustand sein.

Kapitel 5 Prototyp eines SNMP-Agenten

In diesem Kapitel wird es auf die Notwendigkeit und die Erstellung eines SNMP-Agenten mittels Unicenter-SDK eingegangen. Anschließend wird es Unicenter SDK grob beschrieben.

5.1 Beschreibung eines Prototyps

Da Unicenter TNG einige wichtigsten Funktionalitäten wie z.B. das Setzen und die Überwachung von Schwellwerten der MIB-Variablen ohne einen extra zu besorgenden Agent nicht vorsieht, kann Unicenter Software Development Kit (SDK) hier eine große Hilfe sein. Selbst wenn man sich so einen Agent besorgt, können damit nur bestimmte Hosts überwacht werden, das heißt, daß ein HP-Printer nur von einem HP-Agent überwacht werden kann.
Wünschenswert wäre dagegen, daß man die MIB-Variablen auf allen Hosts nur mit *einem einzigen Agent* überwachen kann, ohne die zusätzlichen Agenten für die einzelnen Systeme installieren zu müssen.

Mit der Hilfe von Unicenter TNG **SDK** wurde im Rahmen meines Praktikums ein SNMP-Agent geschrieben, der in der Lage ist, folgendes für eine unbegrenzte Hostanzahl auszuführen :

- SNMP Walk-Befehl
- SNMP Get-Befehl
- Setzen von Schwellwerten über die MIB-Variablen
- Überwachen der Bedingungen über die gesetzten Schwellwerte
- Anzeigen von Error-Message's falls die Schwellwerte überschritten bzw. unterschritten werden
- Anzeigen von Normal-Message's falls die Schwellwerte sich wieder in der gesetzten Grenzen einfinden
- Weiterleiten von Error-Message's an den Leitstand Unicenter TNG und an das Event Management falls die Schwellwerte überschritten bzw. unterschritten werden
- Weiterleiten von Normal-Message's an den Leitstand Unicenter TNG und an das Event Management falls die Schwellwerte sich wieder in der gesetzten Grenzen einfinden

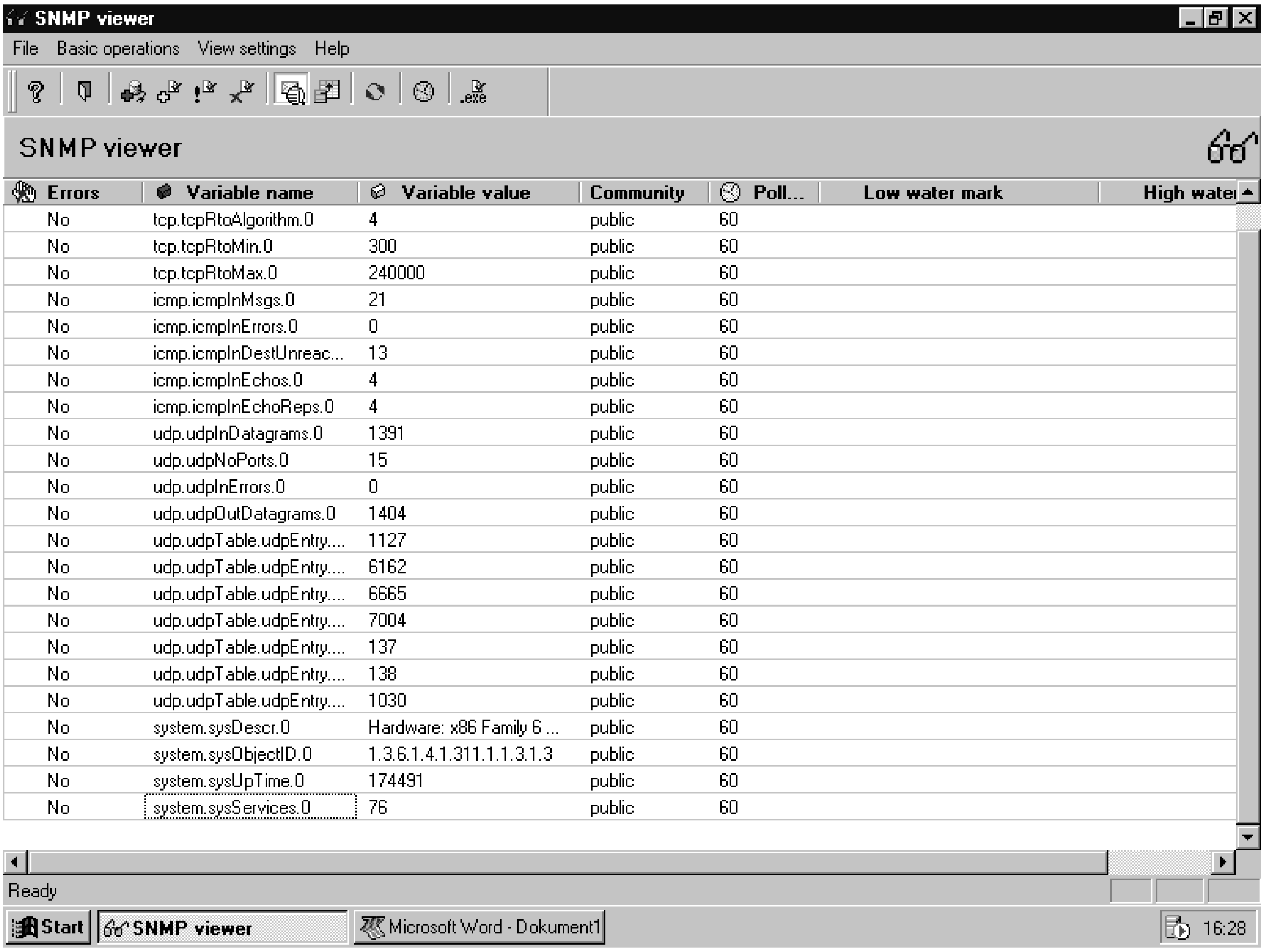


ABBILDUNG 5.1
PROTOTYP EINES SNMP-AGENTEN

5.2 Die SDK- Beschreibung

Mit Unicenter TNG stellt Computer Associates Anwendungsprogrammier-Schnittstellen (Application Programming Interfaces, API's) zur Verfügung, um Zugriff auf die Schichten Enterprise Management und Agenten zu ermöglichen. Diese API's erlauben den Aufruf von Unicenter TNG-Funktionen für die Entwicklung eigener Anwendungen, um die individuellen Anforderungen besser zu erfüllen.

Das SDK umfaßt:
- Agent Factory API
- Enterprise Management API

Mit **Agent Factory API** stellt ein bequemes Verfahren zur Erstellung von Agenten zur Verfügung.
Die Agenten lassen sich in herkömmlichen Client/Server-Umgebungen einsetzen.
Die Agent Factory vereinfacht die Entwicklung von Management-Anwendungen, da nur der Code zur Steuerung von Ressourcen geschrieben werden muß. Die Verwaltungsfunktionen, die die

Mehrheit der herkömmlich entwickelten Agenten umfassen, stehen über die Agent Factory API zur Verfügung.

Die **Enterprise Management API** enthält Funktionen , dank derer die Programme direkt mit den Unicenter TNG-Komponenten für DV-Management integriert werden können.
Es hat mir erlaubt, meinen Agent direkt mit dem Event Management zu verbinden: Error- und Normal-Message's werden an den Leitstand Unicenter TNG geschickt.

Beim Erstellen meines Agenten habe ich z.B. die folgende SDK-Funktionen eingesetzt:

- AWF_AgentExit - signalisiert, daß SNMP-Session beendet ist.
  Syntax: void AWF_AgentExit();
- AWF_AgentReady - Agent ist bereit zum SNMP requests
  Syntax: void AWF_AgentReady()
- AWF_PrintAgent - Printfunktion
  Syntax: int AWF_PrintAgent(FILE *fp);
  Parameter: fp - Pointer auf output file.
- SnmpMgrStrToOid - Konvertierung einer Variable in die enterne Representation
- SnmpMgrOpen - Öffnen eines SNMP-Session
  Syntax: SnmpMgrOpen(strAgent, strCommunity, nTimeout, nRetries)
- EN_CawTo - sendet Normal- und Error-Message zum Unicenter Console

Man kann aus der Erfahrung sagen, daß die SDK-Funktionen leicht in einen Programm zu integrieren, zu implementieren und gut in der Online-Help beschrieben sind.

Kapitel 6 Zusammenfassung

Unicenter TNG bietet eine Lösung, die über das herkömmliche Modell des DV-Managements im Unternehmen hinausgeht und isoliertes System-, Netzwerk- oder Datenbankmanagement in unternehmensweites DV-Management umwandelt - wobei eine Verwaltung aus Sicht der Betriebsabläufe oberstes Ziel ist. Eine solche betriebsablauforientierte Sichtweise stellt in Kombination mit der Architektur von Unicenter TNG einen neuen, leistungsfähigen Ansatz zur Steuerung von IT-Infrastruktur dar.

Unicenter TNG hat mehrere **Vorteile**, unterdessen zum Beispiel

- realitätsgetreue 3-D Oberfläche
- Neugent - unterscheiden zwischen normalen und auffälligen Verhalten im System
- mächtiges SDK - ermöglicht die einfache Erstellung von zusätzlichen Komponenten
- modularer Aufbau - ermöglicht die Anpassung von Unicenter TNG an die Unternehmens Bedürfnisse
- leichte Installationsroutine und Systemanpassung,

aber die **Abwesenheit** wichtiger Funktionalitäten, wie zum Beispiel die Logik in der betriebsablauforientierten Sicht (siehe Kapitel 4) oder die Möglichkeit der Überwachung von Schwellwerten der MIB-Variablen, ohne sich dazu einen zusätzlichen Agent besorgen zu müssen (siehe Kapitel 5), lassen uns annehmen, daß Unicenter TNG noch verbesserungsbedürftig ist. Dies und die Verknüpfung mit Windows NT und die derzeit noch wackelige Anbindung von Microsoft SQL und CA Ingres als bevorzugte relationalen Datenbanken, die als Basis für das objektorientierte Repository dienen, sind wohl die Hauptkritikpunkte an Unicenter TNG. Letzeres wird CA aber spätestens mit der Integration von *Jasmine,* einer objektorientierten Datenbank aus eigenem Haus, beseitigen.

Am Markt positioniert sich Unicenter TNG neben *Tivoli TME10* als dessen direkter Konkurrent in der Klasse der Schwergewichter unter den Systemsmanagementplattformen. Dennoch bleibt es bedingt auch für kleine und mittelgroße Unternehmen (SMEs - Small to Medium-sized Enterprises) interessant, da es dank seines modularen Aufbaus für jedes Unternehmen speziell angepaßt werden kann. Dem Kunde bleibt es überlassen welche Funktion er in seinem System integrieren will, und auf welche er verzichten kann.

Anhang Troubleshooting

Problem	Mögliche Lösung
Beim Start von Node View oder Agent View bekommt man die folgende Fehlermeldung : **„CAE0323: Repository error. Error Code: 22“**	Agent View ist **nicht** installiert, da es *nicht* automatisch während der Unicenter-Installation installiert wird
Beim Start von Node View wird nur Blank Screen ohne weiteren Fehlermeldungen angezeigt	1. Öffnen 2-D Map, klicken mit der rechten Maustaste auf Agent-Icon ; 2. wählen „open details“-Menüpunkt; 3. klicken auf „others tab“; 4. überprüfen **„DSM_server“** - diese Adresse muß korrekt sein und der DSM-Server muß auf dieser Maschine laufen
Im 2-D Map werden dieselben Objekte im Warning- (yellow) und in Agent View im Normal- (green) oder Critical-Zustand (red) angezeigt	NodeView beinhaltet keine Farbe, mit der der Warning-Zustand angezeigt werden kann.
Beim Start von Agent Veiw wird die folgende Fehlermeldung angezeigt : **„coud not connect to orb“**	1. Öffnen 2-D Map, klicken mit der rechten Maustaste auf Agent-Icon ; 2. wählen „open details“-Menüpunkt; 3. klicken auf „others tab“; 4. überprüfen **„DSM_adress“** - diese Adresse muß korrekt sein und das „Common Services“ muß auf dieser Maschine installiert sein
Beim Versuch **„awservices start/stop/status**“ auszuführen, erhält man die folgende Fehlermeldung : **„The dynamic link library mgmtapi.dll coud not be found in the specified path“**	Die Installationsreihenfolge ist nicht korrekt : man muß zuerst das Windows NT Services-Pack und erst danach den SNMP-Dienst installieren.

Abbildungsverzeichnis

Seite 4 Abbildung	1.1	Aufbau von Managementplattform
Seite 7 Abbildung	2.1	Die Architektur von Unicneter TNG
Seite 8 Abbildung	2.2	Node View Window
Seite 10 Abbildung	2.3	System Agent View - Services
Seite 11 Abbildung	2.4	System Agent View - Summary
Seite 13 Abbildung	2.5	Das Polling Modell
Seite 13 Abbildung	2.6	Das Notification Modell
Seite 18 Abbildung	3.1	Agent Technology Architektur
Seite 18 Abbildung	3.2	Agent Schicht
Seite 19 Abbildung	3.3	DSM Schicht
Seite 20 Abbildung	3.4	WorldView Schicht
Seite 23 Abbildung	4.1	Architektur von Business Process View
Seite 25 Abbildung	4.2	Printer BPV
Seite 26 Abbildung	4.3	Beispiel-BPV
Seite 26 Abbildung	4.4	Beispiel- BPV
Seite 29 Abbildung	5.1	Prototyp eines SNMP-Agenten

Literaturverzeichnis

[RiSt1998] „Working with Unicenter TNG“
Rick Sturm, Audrey Rasmussen, Robert H. Taylor
QUE-Verlag 1998

[UnSDK] Unicenter TNG SDK Programmer’s Guide

[ReGu] Reference Guide

[wwwCA] www.cai.com

[wwwUDE] www.unicentershop.de