



Universität Stuttgart

Institut für Nachrichtenvermittlung und Datenverarbeitung

Prof. Dr.-Ing. P. Kühn

54. Bericht über verkehrstheoretische Arbeiten

**WISSENSBASIERTE DIAGNOSE LOKALER NETZE
IM WIRKBETRIEB**

von

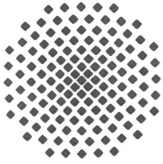
Jürgen M. Schröder

1993

© 1993 Institut für Nachrichtenvermittlung und Datenverarbeitung Universität Stuttgart

Druck: E. Kurz & Co., Druckerei + Reprografie GmbH., Stuttgart

ISBN 3-922403-64-6



University of Stuttgart

Institute of Communications Switching and Data Technics

Prof. Dr.-Ing. P. Kühn

54th Report on Studies in Congestion Theory

**KNOWLEDGE BASED ONLINE DIAGNOSIS
OF LOCAL AREA NETWORKS**

by

Jürgen M. Schröder

1993

Knowledge Based Online Diagnosis of Local Area Networks

Summary

Local Area Networks (LANs) are the "key component" for Computer Integrated Manufacturing (CIM). They interconnect individual computing systems to form large distributed systems. Realized systems for CIM interconnect around thousand - sometimes up to tenthousand individual computing systems as, for example, personal computers, mainframes, workstations, machine controllers, interworking units, etc. LANs function as the backbone of those complex distributed systems. The proper function of the entire system depends highly on the error free operation of the LAN. Due to that the operator of a LAN needs efficient tools which help him to manage the LAN. Since a few years the International Organization for Standardization (ISO) is working on a standard for network management. The standard will cover the following aspects of network management:

- Configuration Management
- Security Management
- Performance Management
- Fault Management
- Accounting Management

This report is concerned with a general concept for online monitoring and fault diagnosis of LANs and an example implementation for a LAN based on the IEEE 802.3 standard, also known by the name "Ethernet". The proposed system is subdivided into the following subsystems:

- Distributed Measurement System
- Monitoring System
- Diagnosis System

In the report the Distributed Measurement System and the Diagnosis System, which is based on the architecture of knowledge based systems, are described in detail.

Chapter 1: Introduction

This chapter provides an overview about LANs and their use in CIM. It also contains an introduction to network management, especially to the network management architecture described by ISO.

Chapter 2: Diagnosis Systems

This chapter starts with an introduction to diagnosis systems. After that the general architecture of knowledge based systems - so called expert systems - is described and an introduction to the possibilities of knowledge description is provided. Also, the basics of different problem solving algorithms are presented. The chapter closes with an overview about existing research projects for network management systems based on knowledge based systems.

Chapter 3: Architecture of the Measurement, Monitoring and Diagnosis System

In the scope of the presented research project a general architecture for measurement, monitoring and diagnosis of LANs has been developed. Its principal architecture is presented in this chapter.

Chapter 4: Distributed Measurement System for Local Area Network based on IEEE 802.3

The architecture and the measurement features of the Distributed Measurement System for LANs based on IEEE 802.3 developed within the project is subject of this chapter. Besides other measurement functions, a measurement method and its realisation is described for online time domain reflection measurement on network segments. The reflection curve of a network segment can be regarded as a *finger print* of that segment.

The possibilities of result interpretation and result processing are discussed. To this an item is the use of feed forward neural networks (perceptrons) for pattern recognition of time domain reflection measurements.

Chapter 5: Monitoring System

The Monitoring System and its architecture is described briefly in that chapter. The Monitoring System is subject of further research projects.

Chapter 6: Diagnosis System

A detailed description of the architecture of the Diagnosis System is given in this chapter. The used problem solving algorithms are described and discussed in detail. For knowledge description, two principal relationship architectures are presented. Also, the objects for detailed knowledge description and their relations are proposed. For diagnosis of large distributed systems it is impossible to store all necessary knowledge in one knowledge base due to consistence problems. For solving that problem a modular knowledge base architecture is proposed. With the presented architecture it is also easy to adapt used, proven knowledge bases to new LAN configurations.

In a further part of the chapter the inference system of the diagnosis system is presented and discussed in detail. Also, a method and its results for the determination of the performance of the inference system of the Diagnosis System are presented.

Chapter 7: Conclusion and Outlook

The results are summarized in this last chapter which is completed by an outlook toward future research and development topics in the field of network management.

Inhaltsverzeichnis

Abkürzungen	5
Begriffserläuterungen	6
Formelzeichen	9
1. Einleitung	12
1.1 Lokale Rechnernetze	12
1.2 Lokale Rechnernetze für die Fertigungsautomatisierung	15
1.3 Netzmanagement	19
1.4 Ziele der Arbeit	23
1.5 Gliederung der Arbeit	24
2. Diagnosesysteme	25
2.1 Einführung	25
2.2 Expertensysteme	27
2.3 Expertensysteme für die Netzdiagnose	31
3. Architektur des Meß-, Monitoring- und Diagnosesystems	33
4. Ein verteiltes Meßsystem für lokale Rechnernetze nach IEEE 802.3	35
4.1 Der Standard IEEE 802.3 für lokale Rechnernetze	35
4.1.1 Geschichte des Standards IEEE 802.3 (Ethernet)	35
4.1.2 Der Standard IEEE 802.3 (Ethernet)	36
4.1.3 Anwendung des IEEE 802.3 Standards in der Fertigungsautomatisierung	41
4.2 Architektur des verteilten Meßsystems	42
4.3 Die Basisbaugruppe	44
4.3.1 Architektur der Basisbaugruppe	44
4.3.2 Meßmöglichkeiten der Basisbaugruppe	45
4.3.2.1 Einteilung der Meßmöglichkeiten	45
4.3.2.2 Tests und Messungen im normalen Betrieb	45
4.3.2.3 Tests und Messungen im Monitor-Betrieb	46
4.3.3 Einsatz der Basisbaugruppe und Meßdatenauswertung	47
4.3.3.1 Allgemeines zum Einsatz der Basisbaugruppe	47
4.3.3.2 Echotest	47
4.3.3.3 Zähl- und Filterfunktionen	47
4.4 Die Durchflußmesserbaugruppe	48

4.4.1	Architektur der Durchflußmesserbaugruppe	48
4.4.2	Meßmöglichkeiten der Durchflußmesserbaugruppe	52
4.4.2.1	Messungen an Stationsanschlüssen	52
4.4.2.2	Messungen an Segmentkopplungen	53
4.4.3	Meßdatenauswertung	57
4.5	Die Reflektometerbaugruppe	58
4.5.1	Meßprinzip und Architektur der Reflektometerbaugruppe	58
4.5.2	Meßmöglichkeiten der Reflektometerbaugruppe	63
4.5.3	Meßdatenauswertung	66
4.5.3.1	Allgemeines zur Meßdatenauswertung	66
4.5.3.2	Mustererkennung durch Bestimmung des Korrelationskoeffizienten	70
4.5.3.3	Mustererkennung mit neuronalen Netzen	72
4.5.4	Weitere Systemerweiterungen	77
4.6	Systemsoftware	77
4.6.1	Betriebssystem	77
4.6.2	Kommunikationssystem	79
4.6.2.1	Anforderungen an das Kommunikationssystem	79
4.6.2.2	Das Low Level Transport Protokoll (LLTP)	80
4.6.3	Anwendungen	83
5.	Das Monitoringsystem	85
6.	Ein Diagnosesystem für die wissensbasierte Diagnose in komplexen Systemen	87
6.1	Architektur des Diagnosesystems	87
6.2	Wissensdarstellung	88
6.2.1	Objekte der Wissensdarstellung	88
6.2.2	Objektrelationen	90
6.2.3	Die Wissensbewertung	93
6.2.3.1	Darstellung der Wissensbewertung	93
6.2.3.2	Operatoren für die Wissensbewertung	94
6.2.3.2.1	Die Negation	94
6.2.3.2.2	Die UND-Verknüpfung	94
6.2.3.2.3	Die ODER-Verknüpfung	96
6.2.3.2.4	Der Mittelwert-Operator	97
6.2.3.2.5	Der kompensatorische UND-Operator	98
6.2.3.2.6	Der kompensatorische ODER-Operator	98
6.2.3.3	Die Bewertung von Regeln	99
6.2.3.4	Die Bewertung von Prämissen	100
6.3	Modulare Wissensbasis	100
6.3.1	Aufbau der Wissensbasis	100

6.3.2 Zusammenbindung der Wissensbasismodule	102
6.4 Die Wissenserwerbskomponente	103
6.4.1 Architektur der Wissenserwerbskomponente	103
6.4.2 Syntax zur Wissensbeschreibung	104
6.5 Die Problemlösungskomponente	110
6.5.1 Architektur der Problemlösungskomponente	110
6.5.2 Agendaoperationen	111
6.5.2.1 Einteilung der Agendaoperationen	111
6.5.2.2 Allgemeine Agendaoperationen	112
6.5.2.3 Agendaoperationen für hierarchische Wissensbasismodule	114
6.5.2.4 Agendaoperationen für heterarchische Wissensbasismodule	114
6.5.3 Parameter zur Problemlösungssteuerung	115
6.5.4 Die Ablaufsteuerung	116
6.5.4.1 Anforderungen an die Ablaufsteuerung	116
6.5.4.2 Die zentrale Ablaufsteuerung	117
6.5.4.3 Die Teilablaufsteuerungen	118
6.5.4.3.1 Gemeinsamkeiten beider Teilablaufsteuerungen	118
6.5.4.3.2 Die hierarchische Ablaufsteuerung	120
6.5.4.3.3 Die heterarchische Ablaufsteuerung	125
6.5.5 Die Monitoringschnittstelle	128
6.5.6 Bewertung der Problemlösungskomponente	131
6.6 Die Erklärungskomponente	134
6.6.1 Die Architektur der Erklärungskomponente	134
6.6.2 Verbindung zwischen Problemlösungskomponente und Erklärungskomponente	135
7. Zusammenfassung und Ausblick	136
7.1 Zusammenfassung	136
7.2 Ausblick	~137
Literaturverzeichnis	138

Abkürzungen

AAP	Application Access Point
ACSE	Association Control Service Element
AI	Artificial Intelligence
AUI	Attachment Unit Interface
CAM	Computer Aided Manufacturing
CIM	Computer Integrated Manufacturing
CMIP	Common Management Information Protocol
CMIS	Common Management Information Service
CRC	Cyclic Redundancy Check
CS	Circuit Switching
CSMA/CD	Carrier Sense Multiple Access / Collision Detection
DAAP	Destination Application Access Point
DECnet	Digital Equipment Corporation Network Architecture
DMS	Data Management System (Northern Telecom)
DNA	Digital Network Architecture
DQDB	Distributed Queue Dual Bus
ECMA	European Computer Manufactures Association
ECL	Emitter Coupled Logic
EPROM	Electrically Programmable Read Only Memory
ETSI	European Telecommunications Standards Institute
EWSD	Elektronisches Wählsystem Digital
FDDI	Fibre Distributed Data Interface
HP	Hewlett Packard
HSLAN	High Speed Local Area Network
IEEE	Institute of Electrical and Electronics Engineers
IBM	International Business Machines Corporation
ISO	International Organization for Standardization
KI	Künstliche Intelligenz
LAN	Local Area Network
LLC	Logical Link Control
LLTP	Low Level Transport Protocol
LE	Layer Entity
LME	Layer Management Entity
MAC	Media Access Control
MAN	Metropolitan Area Networks
MAP	Manufacturing Automation Protocol
MASP	Measurement Application Service Protocol

MAU	Medium Attachment Unit
MIB	Management Information Base
NEC	Nippon Electric Company
OSF	Open Software Foundation
OSI	Open Systems Interconnection
PAL	Programmable Array Logic
PBX	Private Branch Exchange
PC	Personal Computer
PDU	Protocol Data Unit
PS	Packet Switching
QOS	Queue Operating System
RAM	Random Access Memory
REACT	Real-Time Expert Analysis and Control Tool
ROM	Read Only Memory
RTES	Real Time Expert System
SAAP	Source Application Access Point
SMAP	System Management Application Process
SMAE	System Management Application Entity
SMART	Switching Maintenance and Analysis Repair Tool
SNA	Systems Network Architecture
SNMP	Simple Network Management Architecture
SFD	Start-Frame-Delimiter
TAP	Transceiver Access Point
TCP/IP	Transmission Control Protocol / Internet Protocol
UNMA	Unified Network Management Architecture
WAN	Wide Area Network
WBM	Wissensbasismodul
10Base2	Segmentkabel mit 6 dB/185m (Cheapernet)
10Base5	Segmentkabel mit 6 dB/500m (Ethernet)
10BaseT	Twisted Pair Kabel für sternförmige Verkabelung (110m max. vom Verteiler (Hub))

Begriffserläuterungen

Anwendungs-
system Unter diesem Begriff werden die Schichten 5-7 des OSI-Referenzmodells für offenen Kommunikation zusammengefaßt.

Basisbaugruppe	Grundbaugruppe einer Meßstation. Die Minimalkonfiguration einer Meßstation enthält neben der Stromversorgung nur eine Basisbaugruppe. Die Basisbaugruppe wird zur Steuerung der Erweiterungsbaugruppen verwendet.
Basismeßauftrag	Elementarer Meßauftrag, der als ganzes von einer Meßstation durchgeführt werden kann. Eine Meßfunktion kann identisch mit einem Basismeßauftrag sein.
Basisband	Die Übertragung erfolgt direkt über das Übertragungsmedium (ohne Träger-signal). Es steht für die Datenübertragung nur ein Übertragungskanal zur Verfügung.
Breitband	Über ein Übertragungsmedium werden im Frequenzmultiplex mehrere Übertragungskanäle zur Verfügung gestellt.
Carrierband	Über das Übertragungsmedium wird durch Modulation einer Trägerfrequenz übertragen. Es steht nur ein Übertragungskanal zur Verfügung. Carrierbandsysteme sind unempfindlicher gegenüber äußeren Störeinflüssen, wie z.B. starken elektromagnetischen Feldern, als Basisbandsysteme.
Diagnosesystem	System zur Anwenderunterstützung bei der Fehlersuche in Systemen.
Erklärungs-komponente	Teilsystem eines Diagnosesystems. Die Erklärungskomponente erläutert dem Anwender die Vorgehensweise des Diagnosesystems.
Erweiterungs-baugruppen	Baugruppen zur Erhöhung der Funktionalität einer Meßstation. (Reflektometerbaugruppe, Durchflußmesserbaugruppe usw.)
Fallspezifisches Wissen	Vom aktuellen Fehlerfall abhängiges Wissen (z.B. die aktuelle Bewertung von Symptomen).
Heterarchische Modellierung	Der Suchraum für die Problemlösungskomponente kann durch eine netz-artige Struktur von Komponenten und zugehöriger Verbindungen beschrieben werden. Jede Komponente einer solchen Modellierungsebene kann wiederum in ein Netz von Komponenten und zugehörigen Verbindungen verfeinert werden, wobei die Verbindungen der verfeinerten Komponente in die verfeinerte Komponente übernommen werden müssen. Wird eine Komponente im Verlauf einer Problemlösung besonders verdächtig, dann wird deren Verfeinerung für die weitere Diagnose herangezogen.

Hierarchische Modellierung	Die Komponenten zur Wissensbeschreibung werden nach hierarchischen Prinzipien einander zugeordnet. Eine Komponente kann daher nur eine Vorgängerkomponente, aber mehrere Nachfolgerkomponenten haben.
Meßantwort	Antwort einer Meßstation auf einen Basismeßauftrag.
Meßauftrag	Szenario einer Messung am lokalen Rechnernetz. Ein Meßauftrag kann aus mehreren Basismeßaufträgen zusammengesetzt sein. Die zeitliche Bestimmung, wann die Basismeßaufträge gestartet und gestoppt werden, ist Bestandteil des Meßauftrags.
Meßfunktion	Meßfunktion, die von der Basisbaugruppe oder einer Erweiterungsbau- gruppe durchgeführt werden kann. Basismeßaufträge werden durch Meß- funktionen realisiert.
Meßstation	Station ("Sensor") des verteilten Meßsystems. Für die Überwachung und Diagnose des lokalen Rechnernetzes werden im Rechnernetz verteilt Meßstationen installiert. Die Meßstationen werden in ihrer Gesamtheit als verteiltes Meßsystem bezeichnet.
Meßsystem	System zur Meßdatenerfassung. Eine Meßsystem besteht aus Sensoren zur Meßdatenaufnahme.
Meßsystem- Protokoll- Architektur	Architektur des Kommunikationssystems, das für die Kommunikation zwischen Meßstationen und der Monitoring- und Diagnosestation verwendet wird.
Monitoring- und Diagnosestation	Station im lokalen Rechnernetz, auf der das Monitoring- und Diagnosesystem betrieben wird.
Monitoringsystem	Zentrales System zur Steuerung der Meßstationen des Meßsystems und zur Auswertung der Meßdaten.
Monitoringschnitt- stelle	Schnittstelle des Diagnosesystems für die automatische Erfassung von fehlerspezifischem Wissen.
Problemlösungs- komponente	Komponente des Diagnosesystems. Sie steuert die Lösungssuche innerhalb des statischen Wissens der Wissensbasis anhand von aktuellem fallspezi- fischem Wissen. Die Problemlösungskomponente wird auch als Inferenz- komponente oder Inferenzmaschine bezeichnet.
Statisches Wissen	Wissen allgemeiner beschreibender Art. Hier: Wissen, das anhand von Objekten beschrieben in einer Wissensbasis abgelegt wurde.

Transceiver	Realisierung der Medium Attachment Unit (MAU). Zugang zum Übertragungsmedium bei lokalen Rechnernetzen nach IEEE 802.3.
Transportsystem	Unter diesem Begriff werden in lokalen Rechnernetzen die Schichten 1-4 des OSI Referenzmodells für offene Kommunikation zusammengefaßt.
Wissensbasis	Datenbank, in der das Wissen über einen geschlossenen Wissensbereich in einer geeigneten Form für eine Problemlösungskomponente abgelegt wurde.
Wissensbasis-modul	Teil einer Wissensbasis. Eine Wissensbasis enthält mindestens ein Wissensbasismodul.
Wissensdarstellung (Wissensrepräsentation) Methode zur Darstellung von Wissen.	
Wissenserwerbskomponente	Komponente des Diagnosesystems. Sie unterstützt den Anwender bei der Erstellung von Wissensbasismodulen.
Wissensbeschreibungsdateri	Datei, in der das Diagnosewissen mittels einer speziellen Syntax beschrieben wird.

Formelzeichen

a	Korrekturfaktor für die Abtastzeit T_s zur Eliminierung des Jitters der Abtastfrequenz.
$BW(O)$	Bewertungswert des Objektes O
c	Lichtgeschwindigkeit $c \approx 30$ cm/ns
E	Der Energie eines Reflektionssignales proportionale Größe
$f_{abt}(t)$	Abtastfunktion
$F_{abt}(f)$	Fouriertransformierte der Abtastfunktion
G	Mittelwert eines Musterdatensatzes (entspricht dem Gleichanteil des Musterdatensatzes)
$GW(BW(O))$ $= GW(O)$	Gewichtung eines Bewertungswertes. Durch die Gewichtung von Bewertungswerten kann berücksichtigt werden, wie stark die Bewertung eines Objektes zu einer Aussage beiträgt. $GW(O)$ ist die Kurzschreibweise für $GW(BW(O))$.
HW	Häufigkeitswert. Anwendung bei hierarchischen Wissensbasen. Der Häufigkeitswert wird der Relation von Komponente zu Nachfolgerkomponenten zugeordnet. Er beeinflusst, welche der Nachfolgerkomponenten einer Komponente zuerst untersucht wird.
i	Istmusterdatensatz
k	Korrelationskoeffizient

K_{ABS}	Absoluter Kollisionsbetrag an einem Netzsegment
KE	Anzahl der vom Übertragungsmedium empfangenen Datenpakete, bei denen eine Kollision auftrat.
KS	Anzahl der Kollisionen, die beim Senden auftraten.
k_{th}	Schwelle des Korrelationskoeffizienten
l	Werteanzahl eines Teilmusterdatensatzes
l_f	Abstand des Fehlerortes vom Meßort
$L_{Präambel}$	Größe der Präambel eines Datenpaketes in Byte
L_{SFD}	Größe des Start Frame Delimiters in Byte
λ_{oder}	Parameter der kompensatorischen ODER-Verknüpfung
λ_{und}	Parameter der kompensatorischen UND-Verknüpfung
m	Musterwerteanzahl eines Datensatzes einer Reflektionsmessung
m_A	Musterwerteanzahl des aktuell gemessenen Datensatzes
$\max[]$	Maximum-Operator. Realisierung der ODER-Verknüpfung
$\min[]$	Minimum-Operator. Realisierung der UND-Verknüpfung
m_S	Musterwerteanzahl des Sollmusterdatensatzes
MT	Maximaler Timeout einer Menge von Methoden
O	Offset eines Datensatzes einer Reflektionsmessung
PE	Anzahl der vom Übertragungsmedium empfangenen Datenpakete
P_{GES}	Gesamtes Datenverkehrsaufkommen in einem Netzsegment
P_{RAT}	Verhältnis zwischen empfangenen und gesendeten Datenpaketen an einem bestimmten Punkt eines lokalen Rechnernetzes
PS	Anzahl der von der Station zur MAU übertragenen Datenpakete
P_{STAT}	Datenverkehrsbeitrag einer Station
$q(t)$	Fourierrücktransformierte der Rechteckfunktion $Q(f)$
$Q(f)$	Rechteckfunktion im Frequenzbereich (Tiefpaßfilter)
r	Vektor der Dimension m der Abtastwerte r_i einer Reflektionskurve.
$r(t)$	Kontinuierliche Zeitfunktion einer Reflektionskurve
$R(t)$	Fouriertransformierte einer kontinuierlichen Reflektionskurve
$r_d(t)$	Diskrete Zeitfunktion einer Reflektionskurve
$R_d(f)$	Fouriertransformierte einer zeitdiskreten Reflektionskurve
ΔS	Örtliche Auflösung einer Reflektionsmessung
\underline{s}	Sollmusterdatensatz
s_S	Länge eines Netzsegmentes
$TH_{Diagnose}$	Schwellenwert für die Bewertung von Diagnosen. $TH_{Diagnose}$ gibt an, ab welcher Bewertung eine Diagnose akzeptiert werden kann.
$TH_{DynDiag}$	Aktuelle maximale Bewertung einer Diagnose
TH_{Regel}	Schwellenwert für die Bewertung von Symptomen durch Regeln. Nur wenn der Betrag einer Bewertung einer Regel $ TH_{Regel} $ überschreitet, wird das entsprechende Symptom durch eine Regel bewertet.

TH_{Symptom}	Schwelle für die Einbeziehung von Symptomen in die weitere Problemlösung. Nur Symptome, deren Bewertungswert höher als TH_{Symptom} ist, werden berücksichtigt.
t_m	Meßzeit an einem 500 m langen Netzsegment
t_{MAU}	Einfache Signallaufzeit zwischen 2 MAUs an einem 10Base5 Netzsegment
T_p	Maximale Signalübertragungszeit in einem lokalen Rechnernetz nach IEEE 802.3 zwischen 2 Netzpunkten.
T_{PSFD}	Dauer der Datenpaketpräambel und des Start Frame Delimiters
T_S	Zeitintervall zwischen 2 Signalabtastungen
t_s	Einfache Signallaufzeit eines Netzsegmentes mit 500 m Länge
T_w	Zufällige Wartezeit
u	Werteanzahl, um die sich benachbarte Teilmusterdatensätze überlappen
v	Signalausbreitungsgeschwindigkeit auf einem Segmentkabel
/	Negation

1. Einleitung

1.1 Lokale Rechnernetze

Lokale Rechnernetze (Lokale Netze, Local Area Networks, LANs) sind Datennetze, die für eine begrenzte lokale Ausdehnung (Bürogebäude, Fabrikgelände etc.) optimiert sind. Es handelt sich dabei um Datenkommunikationssysteme, die einen Übertragungskanal mit mittlerer bis großer Datenübertragungsrate, kurzen Antwortzeiten und kleinen Fehlerraten zur Verfügung stellen, damit mehrere Endgeräte miteinander kommunizieren können. Charakteristisch ist hierbei, dass für die Vergabe des einzelnen Übertragungskanals ein dezentrales Medienzugriffsverfahren eingesetzt wird. Im Gegensatz hierzu werden in Weitverkehrsnetzen (Wide Area Networks, WANs) Punkt- zu- Punkt- Verbindungen zwischen einzelnen Netzknoten realisiert. Weitverkehrsnetze sind in der Regel öffentliche Netze, die von jedermann genutzt werden können. Lokale Netze werden durch private Organisationen kontrolliert und genutzt. Lokale Netze verschiedener Standorte können über Weitverkehrsnetze miteinander verbunden werden. Bild 1.1 zeigt die Einordnung lokaler Netze nach geographischer Ausdehnung und Übertragungsrate gegenüber anderen Datennetzen.

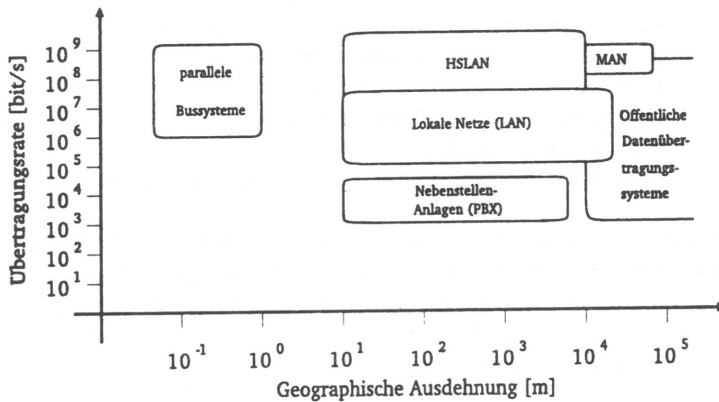


Bild 1.1 Einordnung der lokalen Rechnernetze nach Übertragungsrate und geographischer Ausdehnung gegenüber anderen Datennetzen.

Die Grenzen sind dabei fließend. Aktuell wird an den Entwicklungen von High Speed Local Area Networks (HSLAN) und Metropolitan Area Networks (MAN) gearbeitet. Hauptmerkmal der HSLANs sind Übertragungsraten über 100 Mbit/s. MANs erlauben neben höheren Datenraten auch die Abdeckung größerer geographischer Bereiche. Sie sind dadurch nicht notwendigerweise unter der Kontrolle eines einzigen Betreibers.

Lokale Rechnernetze werden bitseriell betrieben. Die geographische Ausdehnung wird hauptsächlich durch Übertragungstechnische Faktoren wie Dämpfung, Reflektion oder Dispersion begrenzt. Im Gegensatz dazu werden parallele Bussysteme (z.B. Multiprozessorsysteme) zur Überbrückung geringer Distanzen eingesetzt.

Nebenstellenanlagen, vor allem ISDN-fähige, unterscheiden sich von lokalen Rechnernetzen durch unterschiedliche Vermittlungsverfahren. Analoge und ISDN-fähige Nebenstellenanlagen verwenden die Leitungsvermittlung (Circuit Switching CS), die zwischen den beteiligten Endeinrichtungen für die gesamte Verbindungsdauer einen Übertragungsweg zur Verfügung stellt. Lokale Netze arbeiten in aller Regel nach dem Prinzip der Paketvermittlung (Paket Switching PS), da die Vermittlungssteuerung über alle am Netz angeschlossenen Stationen verteilt ist und das vermittelnde Netz nur einen Übertragungskanal besitzt.

Als Einsatzgebiete für lokale Netzwerke können u.a. aufgeführt werden:

- Kommunikation von multifunktionalen Arbeitsplatzsystemen zum schnellen Informationsaustausch zwischen den Abteilungen einer Institution.
- Kommunikation mit zentralen (z.B. innerbetrieblichen) Einrichtungen, um die Aktualität der verwendeten Daten sicherzustellen.
- Kommunikation zwischen unterschiedlichen Endsystemen.
- Kommunikation zwischen Fertigungseinrichtungen in der Fertigungsautomatisierung (Computer Aided Manufacturing, (CAM)).
- Kommunikation zwischen Fertigungseinrichtungen und Fertigungsplanung (Computer Integrated Manufacturing, (CIM)).
- Gemeinsame Benutzung von Betriebsmitteln oder Netzdiensten.

Die Architektur lokaler Rechnernetze kann anhand des Basis-Referenzmodells der International Organization for Standardization (ISO) für die offene Kommunikation (Open Systems Interconnection, OSI) in sieben Schichten gegliedert werden [76]. Bild 1.2 zeigt das OSI Referenzmodell. Jede Schicht erbringt dabei Dienste, die von der darüberliegenden Schicht zur Realisierung ihrer Funktionalität verwendet werden.

Die Funktionalitäten der sieben Schichten des OSI- Referenzmodells können in zwei Aufgabengebiete unterteilt werden. Die Schichten 1 bis 4 erbringen den gesicherten Nachrichtentransport, die Schichten 5 bis 7 enthalten die anwendungsorientierten Funktionen. Man bezeichnet deshalb diese Schichten auch als Anwendungssystem. Aufgaben des Anwendungssystems sind u. a.:

- Dialogsteuerung (*Kommunikationssteuerungsschicht* (Session Layer), Schicht 5),
- Datendarstellungsauswahl und ggf. die Datentransformation (*Darstellungsschicht* (Presentation Layer), Schicht 6),
- Identifikation, Autorisierung und die Vereinbarung von Anwendungsklassen (*Verarbeitungsschicht* (Application Layer), Schicht 7).

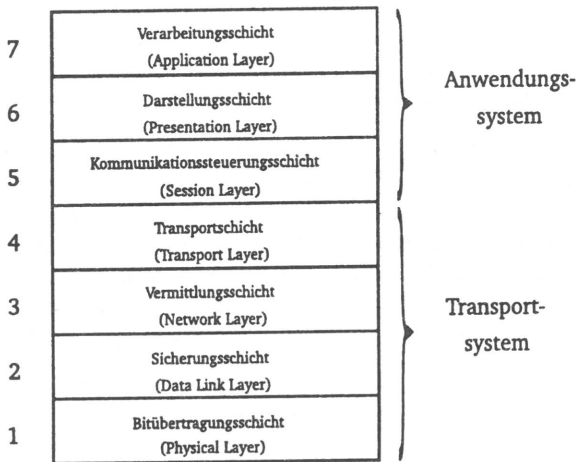


Bild 1.2 Das OSI-Referenzmodell für die Kommunikation offener Systeme

Die Funktionen des Anwendungssystems bauen auf den Transportdienst auf. Sie sind vollständig entkoppelt davon, wie der Transportdienst erbracht wird. Die unteren 4 Schichten des OSI-Referenzmodells werden deshalb auch unter dem Begriff Transportsystem zusammengefaßt. Für die Einordnung lokaler Rechnernetze in das OSI-Referenzmodell sind nur die Schichten des Transportsystems von Bedeutung.

In der *Bitübertragungsschicht* (Physical Layer) werden die elektrischen, mechanischen, aber auch die funktionellen Eigenschaften festgelegt, die zur reinen Bitübertragung über einen ungesicherten Nachrichtenkanal notwendig sind.

Die *Sicherungsschicht* (Data Link Layer) erbringt eine gesicherte Nachrichtenübertragung auf einer Teilstrecke einer Verbindung. Weiter ist in Schicht 2 die Möglichkeit einer Datenflußsteuerung mit Reihenfolgesicherung vorgesehen (wird in der Regel nur bei verbindungsorientierten Systemen verwendet). Ebenso gehören zu dieser Schicht Funktionen zum Verbindungsaufbau und -abbau einer Schicht 2 Verbindung.

Die *Vermittlungsschicht* (Network Layer) fügt einzelne gesicherte Teilverbindungen zur einer Verbindung zwischen 2 Endsystemen zusammen. Weiter werden von der Vermittlungsschicht Dienste für den Verbindungsaufbau und -abbau zwischen Endsystemen, die Verkehrslenkung (Routing) und das Multiplexen mehrerer Verbindungen zwischen Endsystemen erbracht.

Die *Transportschicht* (Transport Layer) führt die Abbildung der Endsystemverbindungen auf die Kommunikationsinstanzen durch, die durch das Anwendungssystem definiert werden. In der Transportschicht können Teilnehmerverbindungen auf mehrere Endsystemverbindungen aufge-

spalten werden (Splitting). Die Datenflußsteuerung und die Fehlerkontrolle über das gesamte Transportsystem hinweg wird in der Transportschicht durchgeführt. Die Instanzen der Transportschicht existieren nur in den Endsystemen und stellen einen netzunabhängigen Datentransport für das Anwendungssystem bereit.

Eine Einordnung lokaler Netze in die OSI-Architektur macht eine Unterteilung der Sicherungsschicht sinnvoll. Der Zugang zum eigentlichen Übertragungsmedium muß in jeder Station bereitgestellt werden. Dadurch sind die Schichten 1 und 2 abhängig von der verwendeten LAN-Technologie. Aus diesem Grund wurde die Sicherungsschicht in zwei Teilschichten (2a, 2b) aufgeteilt, die Kanalzugriffsschicht (Media Access Control Sublayer, MAC) und die logische Sicherungsschicht (Logical Link Control Sublayer, LLC). Durch die Aufteilung der Schicht 2 in Teilschichten wird eine Abtrennung des technologieabhängigen Teils der lokalen Netze möglich.

Die Architektur von lokalen Netzen unterscheidet sich nur bis zur Schicht 2a MAC. Bild 1.3 zeigt den Stand der LAN-Normung [69, 70, 71, 72, 81, 82, 83, 84].

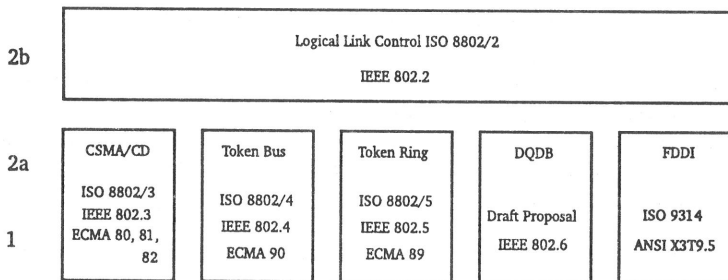


Bild 1.3 Stand der LAN- und MAN-Normung

Lokale Rechnernetze (Local Area Networks, LANs) werden heute zunehmend in der Fertigungs- und in der Büroautomatisierung eingesetzt, um eine möglichst flexible und schnelle Informationsübertragung zu gewährleisten. Sie bilden damit das Rückgrat komplexer verteilter Systeme, in denen unterschiedlichste Geräte und Anwendungen miteinander im Verbund betrieben werden. Durch den Einsatz lokaler Rechnernetze entstehen Gesamtsysteme, deren Funktionalität von entscheidender Bedeutung für ein Unternehmen sein kann [191].

1.2 Lokale Rechnernetze für die Fertigungsautomatisierung

An lokale Rechnernetze für die Fertigungsautomatisierung (Computer Integrated Manufacturing, CIM) müssen, gegenüber lokalen Rechnernetzen für die Büroautomatisierung, zusätzliche Anforderungen gestellt werden [117, 118, 177]. Die wesentlichen zusätzlichen Anforderungen sind:

- Echtzeit (Realzeit) Eigenschaften für die Übertragung zeitkritischer Daten (Besonders in Bereichen, in denen Sensoren und Aktoren eingesetzt werden, Feldebene).
- Große Störsicherheit. Einzelne Komponenten werden häufig in Umgebungen mit starken elektrischen Feldern, hohen Temperaturen und Schmutzintensitäten installiert.
- Integrationsfähigkeit. Die typischen Geräteklassen für die Fertigung (z.B.: Werkzeugmaschinen, speicherprogrammierbare Steuerungen), Fertigungsplanungssysteme (z.B. Mainframes für die Datenverwaltung, CAD-Systeme für die Entwicklung) usw. müssen zu einem Gesamtsystem integriert werden können.

Die Kommunikation innerhalb eines Unternehmens wird daher in Ebenen eingeteilt. Bild 1.4 zeigt Kommunikationshierarchie in einem lokalen Rechnernetz für die Fertigungsautomatisierung [115, 117, 177].

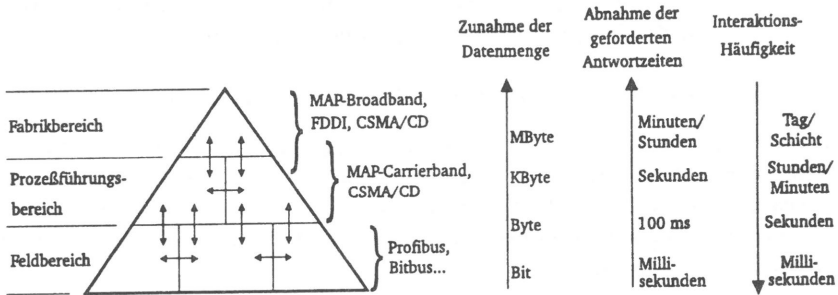


Bild 1.4 Kommunikationshierarchie in einem lokalen Rechnernetz für die Fertigungsautomatisierung

Die unterste Ebene der Kommunikationshierarchie bildet der sogenannte *Feldebereich*. Der *Feldebereich* wird weiter unterteilt in sogenannte *Fertigungszellen*. Im *Feldebereich* werden Maschinen gesteuert. Es fallen direkt maschinenbezogene Daten (Temperaturen, Stückzahlen usw.) an. Die einzelnen *Fertigungszellen* des Feldebereiches kommunizieren untereinander und mit dem *Prozessführungsbereich*.

Im Feldebereich werden nur geringe Datenmengen (Bit) übertragen. Die erforderlichen Antwortzeiten liegen im Millisekundenbereich. Für die Schichten 1 und 2a werden im Feldebereich unterschiedliche Realisierungen verwendet:

- Feldbussysteme (Profibus (DIN 19245) [10], Bitbus, CAN-Bus, DIN-Meßbus u.a.) [115, 116, 184]. Einsatz für die Kommunikation zwischen Steuerungen, Sensoren, Antrieben usw.

- Token Bus Systeme, Carrierbandsysteme [71, 129]. Diese Systeme entsprechen dem Manufacturing Automation Protocol (MAP) Standard 3.0. Einsatz für die Kommunikation zwischen Fertigungszellen, Steuerungen, Leitständen usw.

Dem *Feldbereich* übergeordnet ist der *Prozeßführungsbereich*. Aufgabe des Prozeßführungsbereiches ist die Steuerung und Regelung einzelner *Fertigungszellen* sowie deren produktions-spezifische Einstellung. Zwischen *Prozeßführungsbereich* und *Feldbereich* werden z. B. Soll-daten, NC-Programme usw. ausgetauscht. Die zu übertragenden Datenmengen liegen hier im Bereich von wenigen Byte bis zu einigen KByte. Die erforderlichen Antwortzeiten liegen im Sekundenbereich. Die Schichten 1 und 2a des *Prozeßführungsbereiches* werden realisiert durch:

- Token Bus Systeme, Carrier- und Breitbandsysteme entsprechend dem MAP Standard 3.0. Einsatz für die Kommunikation zwischen Leitständen, Fertigungszellen, Produktionsleitung usw.
- CSMA/CD basierte Systeme. Einsatz für die Kommunikation zwischen Leitständen, Fertigungszellen, Produktionsleitung usw.

Im *Prozeßführungsbereich* sind typischerweise die *Leitwarten* angeordnet.

Dem *Prozeßführungsbereich* ist der *Fabrikbereich* übergeordnet. Der *Fabrikbereich* kann in die Bereiche *Produktionsleitung* und *Unternehmensleitung* unterteilt werden. Die *Produktionsleitung* überwacht und steuert den *Prozeßführungsbereich*. Diese Ebene ist für die Überwachung und Steuerung der gesamten Produktion zuständig. Wesentliche Aufgabe der *Produktionsleitung* ist die *Arbeitsvorbereitung*.

Im Bereich der *Unternehmensleitung* werden Produktionsdaten erfaßt und strategische bzw. administrative Überlegungen in die *Produktionsleitung* eingebracht. Die zu übertragenden Datenmengen im *Fabrikbereich* liegen im Bereich von MByte. Die erforderlichen Antwortzeiten im Minuten- bis Stundenbereich.

Die Teilnetze der einzelnen Ebenen der Kommunikationshierarchie werden durch geeignete Netzkoppeleinheiten (Router, Bridges, evtl. Gateways) [10] miteinander verbunden.

Die Realzeitanforderungen an die Kommunikationsarchitektur werden immer geringer, je weiter man sich in der Kommunikationshierarchie vom *Feldbereich* entfernt. Deshalb nimmt in den höheren Ebenen der Kommunikationsarchitektur die Realisierung der Schichten 1 und 2a, basierend auf CSMA/CD-Systemen (Ethernet) [118], zu. Vielfach werden heute im *Fabrikbereich* auch schnelle Hintergrundnetze (Backbone-Netze) z.B. Fibre Distributed Data Interface (FDDI, 100 Mbit/s) eingesetzt.

Standard für die Kommunikation in Fertigungsautomatisierungsanlagen ist das Manufacturing Application Protocol (MAP) [129]. Dieses Protokoll wurde Anfang der 80er Jahre durch General Motors konzipiert. Auf der National Computing Conference (NCC) 1984 wurde eine Testinstallation von MAP der Öffentlichkeit vorgeführt. MAP stellt ein sogenanntes Protokoll-Profil dar.

Aus einer Reihe von standardisierten Protokollen werden für die Fertigungsautomatisierung geeignete Protokolle für die MAP-Hierarchie zusammengestellt. Seit 1987 liegt die Version 3.0 des Standards vor. Die Version 3.0 wurde für 6 Jahre festgeschrieben, um einen stabilen Zustand für diesen Zeitraum zu gewährleisten.

MAP basiert auf einem Breitbandsystem mit einer Übertragungsrate von 10 Mbit/s in einem 12 MHz Frequenzbereich. Es werden 3 Übertragungskanäle im Frequenzmultiplex bereitgestellt. Ab Version 3.0 ist auch der Einsatz von Carrierband-Teilnetzen mit einer Übertragungsrate von 5 Mbit/s vorgesehen.

Als Medienzugangsverfahren wird das Token Bus Protokoll [71] verwendet. Das Token Bus Protokoll gestattet einen konfliktfreien, deterministischen Zugang zum Übertragungsmedium. Prioritäten und "Immediate Response" sind vorgesehen.

Auf die Schichten 1 und 2 folgt eine leere Vermittlungsschicht (inactivity subset). Darauf setzt eine Transport-Schicht der Klasse 4 auf, deren wesentliche Eigenschaften neben umfangreichen Fehlerentdeckungs- und Behebungsfähigkeiten, verbindungsorientierte Dienste für Flußkontrolle, Segmentierung und Reihenfolgeerhaltung sind.

Die Protokolle des Anwendungssystems bedienen sich der Dienste des Transportsystems. Hierzu zählen:

- Interprozeßkommunikation und -steuerung (Association Control Service Element, ACSE) [79, 80]
- Dateitransfer (File Transfer, Access and Management, FTAM [78])
- Kommandosprache zur Ansteuerung von fertigungsnahen Geräten [85] (Manufacturing Message Specification, MMS)
- Netzmanagement (CMIS/CMIP [94, 95])
- Directory Service [86-93]

Die Standards der genannten Protokolle des Anwendungssystems besitzen weitgehend den Status Draft International Standard oder International Standard und sind damit überwiegend stabil. In den Teilbereich Netzmanagement von MAP wurde in der Version 3.0 kein Standard der ISO übernommen.

Eine Erweiterung des MAP Standard 3.0 wird diskutiert. Es wird die Übernahme der OSI-Netzmanagement-Architektur, einer Netzwerkschicht mit höherer Funktionalität und des CSMA/CD-Medienzugangsprotokolls für die Schichten 1 und 2a erwogen, um nur einige Diskussionspunkte zu nennen.

Zum momentanen Zeitpunkt wird in der Kommunikationshierarchie eines Unternehmens eine Vielzahl unterschiedlicher Protokolle eingesetzt (z.B.: MAP, TCP/IP, DECnet, SNA). Die Ursache für die Vielzahl an Protokollen liegt darin, daß Protokolle, die dem OSI-Standard entsprechen, nur in geringer Anzahl als Produkt zur Verfügung stehen. Die meisten Produkte mit

standardisierter Protokoll Architektur sind für das Manufacturing Automation Protocol (MAP) verfügbar. Langfristig ist abzusehen, daß sich die Vielzahl unterschiedlicher Protokolle zugunsten von OSI-Protokollen verringern wird. OSI-Protokolle bieten eine höhere Funktionalität und Sicherheit, benötigen aber auch mehr Rechenleistung.

1.3 Netzmanagement

Für die Verwaltung großer Rechnernetze, bestehend aus Komponenten unterschiedlicher Hersteller [20, 110, 111, 112, 159, 182], bedarf es eines internationalen Standards. Die ISO ist bemüht, dem Standard für die offene Kommunikation, einen Netzmanagement Standard hinzuzufügen [9, 38, 64, 98-105, 158, 173]. Der Ansatz des ISO Open System Interconnection Network Management deckt folgende Teilbereiche des Netzmanagements ab:

- **Configuration Management**
Zum Configuration Management zählen:
 - Beschreibung der Konfiguration des Netzwerkes
 - Benennung von Betriebsmitteln (Ressourcen)
 - Initialisierung und Stilllegung von Betriebsmitteln
 - Zustandserfassung einzelner Betriebsmittel bei Normalbetrieb und bei Änderungen
 - Erfassung von Konfigurationsänderungen des Netzwerkes
- **Security Management**
Sicherung des Netzes vor unberechtigter Benutzung von Betriebsmitteln. Hierzu zählen:
 - Überprüfung der Teilnehmeridentität
 - Kontrolle und Verwaltung von Berechtigungen
 - Verwaltung von Kryptographieschlüsseln
 - Aufzeichnung und Auswertung von Sicherheitsprotokollen
- **Performance Management**
Überwachung der Leistungsdaten des lokalen Rechnernetzes. Hierzu zählen:
 - Sammeln statistischer Daten (z.B.: Durchsatz, Fehlerrate, Antwortzeit, Verfügbarkeit usw..)
 - Aufzeichnung und Auswertung von Protokollen über die Netzzustandsgeschichte
- **Fault Management**
Hierunter werden alle Funktionen zur Fehlererkennung, Fehlerisolation und Fehlerbehebung zusammengefaßt. Zu den Funktionen gehören:
 - Empfangen von Fehlermitteilungen und Durchführung entsprechender Reaktionen
 - Führen und Auswerten von Fehleraufzeichnungen
 - Durchführung von Tests für die Diagnose

- **Accounting Management**

Erhebung von Gebühren für die Benutzung von Betriebsmitteln. Hierzu zählen die Funktionen:

- Setzen und Überwachen von Schranken für die Benutzung von Betriebsmitteln
- Zusammenfassung der Gebühren für die Benutzung mehrerer Betriebsmittel durch einen Benutzer

Die Ziele von Management-Operationen werden allgemein als *Managed Objects* bezeichnet. Sie stellen die für das Netzmanagement notwendigen Informationen über die Betriebsmittel des Netzes dar. Elemente eines *Managed Objects* sind:

- **Attribute**

Attribute sind Informationselemente, die den Status eines Betriebsmittels repräsentieren oder für das Management relevant sind und von diesem verwaltet werden. Sie können eine oder mehrere Variablen beinhalten. Hierzu zählen:

- Zähler und Pegelwerte, die durch den Betrieb verändert werden,
- Namensparameter, Konfigurationsparameter, Grenzwertparameter, usw.

- **Operationen (Actions)**

Für ein *Managed Object* definierte Operationen beziehen sich auf das, dem *Managed Object* zugeordnete, Betriebsmittel. Hierzu zählen:

- Test mit einem bestimmten Betriebsmittel (z.B. Echofunktionen),
- Aufforderung an ein System, sich selbst zurückzusetzen.

- **Meldungen (Events)**

Vordefinierte Ereignismeldungen, die bei Auftreten eines Ereignisses an den Managerprozeß gesendet werden. Hierzu zählen:

- Überschreiten von Schwellwerten,
- Zustandsänderungen,
- Abschluß einer Operation.

- **Managed Objects**

Weitere *Managed Objects*, die das aktuelle *Managed Object* verfeinern. Die dadurch entstehende hierarchische Struktur wird auch für die Adressierung der *Managed Objects* verwendet.

Alle *Managed Objects* eines Systems werden unter dem Begriff *Management Information Base (MIB)* zusammengefaßt. Die MIB kann als verteilte Datenbank mit verteilten Informationselementen auf den im Netz verteilten Systemen betrachtet werden.

Über die Implementierung der *Managed Objects* wird im Standard keine Aussage gemacht (z.B.: Hardware, Firmware, Software-Variable...). Die *Managed Objects* werden nur auf abstrakter Ebene beschrieben. Ebenfalls wird der Zugriff auf die *Managed Objects* nicht durch den Standard festgelegt. Das OSI-Netzmanagement standardisiert nur die *Managed Objects*, die für die

OSI-Kommunikation relevant sind. Weitere *Managed Objects* können durch den Anwender definiert werden.

Für den Austausch von Management-Daten zwischen verschiedenen System werden vom OSI-Management drei Mechanismen (Bild 1.5) vorgesehen:

- (N)-Layer Operation, (N = 1... 7)
- (N)-Layer Management,
- System Management.

Die *(N)-Layer Operation* verwendet die regulären N-Layer Protokollelemente für die Überwachung einer einzelnen Instanz der Schicht N. Kommunikationspartner sind (N)-Instanzen (Layer Entities, LE). Beispiele hierfür sind: Das Rücksetzen von Verbindungen (RESET in X.25), das gegenseitige Mitteilen von Fehlern zwischen Kommunikationspartnern, Protokollelemente zum Test der Partnerstation (Echotest auf Schicht 2).

Das *(N)-Layer Management* dient der Überwachung und Kontrolle der *Managed Objects* einer Schicht des OSI-Referenzmodells. Diese Aufgaben können in jeder Schicht von einer (N)-Layer Management Instanz (Layer Management Entity, LME) übernommen werden. Die LMEs derselben Schicht kommunizieren untereinander mittels des (N)-Layer-Management Protokolls. Der Informationsaustausch bleibt jedoch auf Management-Daten derselben Schicht beschränkt. Der Austausch von Routing Information zwischen Vermittlungsschichten oder Ladeprotokolle der Schicht 2 zur Initialisierung einer Station sind Beispiele für Layer-Management Protokolle.

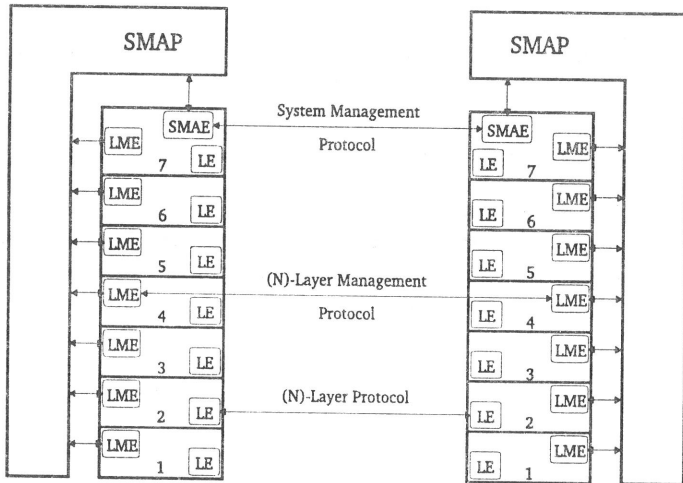


Bild 1.5

Kommunikationsmodell des OSI-Managements

Die Steuerung und Überwachung aller Managed Objects eines Systems ist Aufgabe des *System Management*. Es stellt die einzige Möglichkeit dar, zugleich auf Objekte verschiedener Schichten zuzugreifen. Kommunikationspartner sind hier Anwendungsprozesse (System Management Application Processes SMAPs), die für die Kommunikation Dienste von Instanzen der Anwendungsschicht verwenden (System Management Application Entities, SMAE), die miteinander über System Management Protokolle (Common Management Information Protocol (CMIP), Common Management Information Service (CMS)) [94, 95] kommunizieren.

Voraussetzung für das *System Management* ist, daß alle sieben Schichten des OSI-Referenzmodells zur Verfügung stehen. Deshalb muß der unvollständige Protokollaufbau mancher Systeme (Router, Bridges, Repeater...) erweitert werden, um eine minimale Funktion für das Management bereitzustellen. Dies kann auch durch Auslagerung der höheren Protokollschichten auf andere Stationen mit vollständigem Protokollaufbau erfolgen.

Die Managementfunktionen werden durch mehrere Anwendungsprozesse auf verschiedenen Systemen realisiert. Man unterscheidet zwischen Manager- und Agentprozessen (Manager Process, Agent Process). Management Aktivitäten gehen von den Managerprozessen aus. Sie übernehmen die Verantwortung für Netzteilbereiche (Management Domains) oder das gesamte Netz. Die Managerprozesse kommunizieren mit den Agentprozessen, um Informationen über das Netz zu erhalten, bzw. zu verändern. Auf jeder Station im Netz befindet sich ein Agentprozeß, der für die Managed Objects der Station verantwortlich ist. Er erhält Aufträge vom zuständigen Managerprozeß und ist für die Umsetzung der Aufträge verantwortlich. Bild 1.6 zeigt einen Netzausschnitt mit einem Managerprozeß und 2 Agentprozessen.

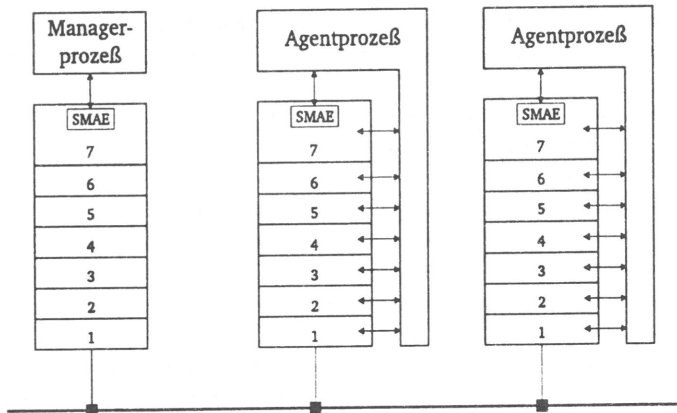


Bild 1.6 Managerprozeß - Agentprozesse.

Der OSI-Netzmanagement Standard beinhaltet keine Beschreibung eines vollständigen Netzmanagement-Systems. Durch den Einsatz von OSI-Netzmanagement werden deshalb nicht alle Netzmanagement-Probleme gelöst [20, 64]. Die eigentliche "Intelligenz" des Netzmanagement

Systems ist nicht Gegenstand des Standards. Ziel der Standardisierungsbemühungen ist die Vereinheitlichung der Repräsentation der Management-Daten und der entsprechenden Zugriffsvorgahren, um die Interoperabilität zwischen Netzmanagement Produkten verschiedener Hersteller zu ermöglichen. Durch die Standards wird letztendlich nur festgelegt, wie ein Ereignis dargestellt und übertragen wird, jedoch nicht, wie darauf reagiert werden soll. Daraus ergibt sich für Hersteller die Freiheit, eigene Netzmanagement Produkte auf der Basis des OSI-Netzmanagements anzubieten. Die Bandbreite von Netzmanagement Produkten wird von einfachen Abfragesystemen, über komfortable Systeme mit aufwendiger Benutzeroberfläche, bis hin zu wissensbasierten Regelsystemen reichen [44, 61, 158].

Bislang werden Produkte für spezifische Netze angeboten, deren Migration zum OSI-Netzmanagement wird jedoch angekündigt (Digital Equipment Corporation (DEC): Enterprise Management Architecture (EMA) [103, 180]; Hewlett Packard (HP): Open View [64]; International Business Machines Corporation (IBM): Netview [64, 188]; AT&T: Unified Network Management Architecture (UNMA) [26, 64, 104]; British Telecom: Concert [189]).

1.4 Ziele der Arbeit

Folgende Aspekte des Netzmanagement sind Ziele dieser Arbeit:

- Die Erfassung von charakteristischen Systemdaten eines lokalen Rechnernetzes für die Betriebsüberwachung und die Diagnoseunterstützung durch ein geeignetes verteiltes Meßsystem.
- Die Auswertung der charakteristischen Systemdaten durch ein zentrales System unter dem besonderen Aspekt der Diagnoseunterstützung und der Vermeidung von Netzausfällen.
- Die Unterstützung des Netzbetreibers bei der Fehlerdiagnose in einem lokalen Rechnernetz durch ein wissensbasiertes System.
- Die Anpaßbarkeit vorhandener, in wissensbasierter Form dargestellter, Fehlerszenarien und der zugehörigen Diagnosen auf unterschiedliche Systemkonfigurationen.
- Die Behandlung großer Wissensbereiche in wissensbasierten Systemen.
- Die Bereitstellung eines wissensbasierten Systems für die Fehlerdiagnose von lokalen Rechnernetzen für die Fertigungsautomatisierung mit direkter Erfassung charakteristischer Systemdaten.

Der Schwerpunkt dieser Arbeit liegt in der Entwicklung einer geeigneten Systemarchitektur für Überwachung und Fehlerdiagnose von lokalen Rechnernetzen.

1.5 Gliederung der Arbeit

In der Einleitung wurde in die Themenbereiche lokale Rechnernetze für die Fertigungsautomatisierung und Netzmanagement eingeführt. Kapitel 2 soll in den Themenbereich der Diagnosesysteme einführen. Dabei wird der potentielle Anwendungsbereich des Network Management berücksichtigt.

In Kapitel 3 wird die Systemarchitektur des Meß-, Monitoring-, und Diagnosesystems vorgestellt, das im Rahmen der Arbeit realisiert wurde. Ziel des Kapitels ist es, einen Überblick über die Gesamtarchitektur des Systems zu vermitteln.

Auf Architektur und die Meßmöglichkeiten des verteilten Meßsystem für lokale Rechnernetze nach IEEE 802.3 wird in Kapitel 4 eingegangen. Es wird neben anderen Baugruppen des Meßsystems eine Reflektometerbaugruppe vorgestellt, welche die Ermittlung von Reflektionskurven an Netzsegmenten nach IEEE 802.3 (Ethernet) 10Base5/10Base2 bei Netzbetrieb zuläßt. Die Meßfunktionen der einzelnen Baugruppen werden vorgestellt und die Auswertemöglichkeiten für Meßergebnisse diskutiert. Hierzu zählt auch der Einsatz nicht zurückgekoppelter neuronaler Netze (Perceptron) für die Mustererkennung bei Reflektionsmessungen.

Kapitel 5 erläutert die Architektur des Monitoringsystems. Hier soll nur ein funktioneller Überblick über das Monitoringsystem gegeben werden. Für einen detaillierten Einblick wird an den entsprechenden Stellen auf vertiefende Literatur verwiesen.

Die Architektur des Diagnosesystems wird ausführlich in Kapitel 6 diskutiert. In den Unterabschnitten des Kapitels wird unter anderem die Problemlösung, die Wissensbeschreibung und der Aufbau der modularen Wissensbasis vorgestellt. Die Leistungsfähigkeit des Diagnosesystems wird abschließend diskutiert.

Kapitel 7 faßt den Inhalt der Arbeit zusammen und gibt einen Ausblick auf weitere Arbeiten.

2. Diagnosesysteme

2.1 Einführung

Der Betrieb komplexer technischer Systeme (z. B. flexible Fertigungssysteme oder Produktionsanlagen) erfordert die kontinuierliche Überwachung und Interpretation einer erheblichen Anzahl von Systemdaten in einer zentralen übergeordneten Stelle (Leitwarte) [22, 65]. Für die Gewährleistung von Sicherheit, Verfügbarkeit und damit der Wirtschaftlichkeit des Systems sind folgende Punkte von Bedeutung:

- Frühzeitiges Erkennen von Anomalien im Systemverhalten
- Richtiges Erkennen des Systemzustandes bei Störfällen
- Schnelles und gezieltes Durchführen von Gegenmaßnahmen

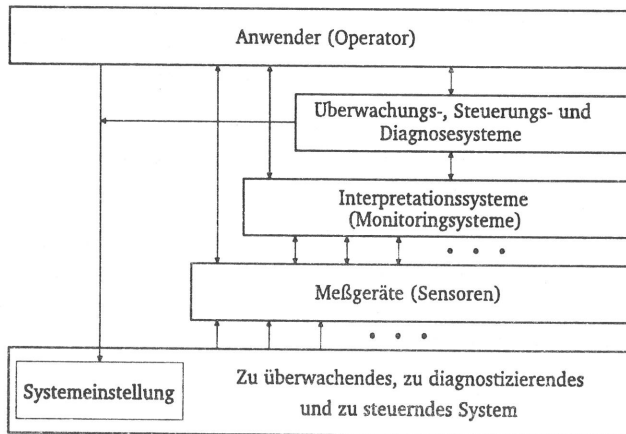


Bild 2.1 Prinzipieller Aufbau von Systemen zur Überwachung, Steuerung und Diagnose von komplexen Systemen.

Bild 2.1 zeigt den prinzipiellen Aufbau von Systemen zur Überwachung, Diagnose und Steuerung von Systemen. Dem zu überwachenden und zu steuernden System werden mit Meßgeräten (Sensoren) Systemdaten entnommen. Bei einfachen Systemen können diese durch den Anwender (Operator) direkt interpretiert und entsprechende Systemeinstellungen, bzw. Systemdiagnosen abgeleitet werden. Mit zunehmender Komplexität wird dies für einen menschlichen Anwender immer schwieriger. Zur Interpretation von Meßdaten werden daher Interpretationssysteme (Monitoringsysteme) verwendet. Solche Systeme bündeln bzw. vorverarbeiten Meßdaten und führen dadurch eine Datenreduktion durch. Durch die Unterstützung von

Interpretationssystemen wird es möglich, komplexere Systeme zu überwachen, zu steuern und zu diagnostizieren.

Für die Automatisierung der genannten Funktionen werden den Interpretationssystemen Überwachungs-, Steuerungs- und Diagnosesysteme übergeordnet. Diese übergeordneten Instanzen erzeugen aus den Informationen, die durch die Interpretationssysteme bereitgestellt werden, Überwachungs-, Steuer- oder Diagnosedaten. Hierfür werden neben bekannten Systemen der Regelungstechnik [120] auch wissensbasierte Systeme eingesetzt [45, 55, 56, 62, 165].

Die Anwendung wissensbasierter Systeme ist vor allem dann sinnvoll, wenn die erforderliche Vorgehensweise nicht mehr in Algorithmen ausgedrückt werden kann. Dies gilt insbesondere für den Bereich der Diagnose. Diagnoseexperten gehen bei der Fehlersuche in komplexen Systemen in der Regel heuristisch vor. Zudem wird hier häufig unsicheres Wissen [148, 156, 197] in den Entscheidungsprozeß miteinbezogen.

Für den Anwendungsbereich der lokalen Rechnernetze werden auf dem Markt verschiedene Geräte angeboten, die zum Meßgerätebereich gezählt werden können. Hierzu zählen:

- Protokollanalytoren (Hewlett Packard: HP4972A, Siemens: B5100, Wandel & Goltermann: DA30, Network General: Sniffer, um nur einige Beispiele zu nennen)
- Verteilte Systeme (R&K: Ethernet Box, Hewlett Packard: LAN-Probe)

Die angebotenen Protokollanalytoren sind in der Regel heute noch nicht fernsteuerbar, d.h. es handelt sich um stand-alone Systeme, die sich nicht in eine Gesamtsystemarchitektur nach Bild 2.1 einbinden lassen. Seit kurzem wird von der Firma Hewlett Packard ein System (LAN-Probe) mit verteilten Meßkomponenten und zentraler Verwaltung für Ethernet-Netzwerke angeboten. Das System erlaubt die Erfassung von Rechnernetzdaten der untersten Schichten des OSI-Referenzmodells und kann bislang nicht um ein Meßsystem für die Analyse von Protokollen höherer Schichten erweitert werden. Nach dem Architekturmodell in Bild 2.1 umfaßt das System den Meßsystem- und den Interpretationssystembereich.

Für Telekommunikationsnetze existiert eine Vielzahl von fernsteuerbaren Meßgeräten verschiedener Hersteller (Anritsu, Hewlett Packard, Tektronix, Wandel & Goltermann usw.). Für die zentrale Erfassung und Interpretation der Systemdaten werden von den Telefongesellschaften eigene, oder im Auftrag entwickelte, Interpretationssysteme eingesetzt. Im Telekommunikationsbereich werden schon übergeordnete Überwachungs-, Steuerungs- und Diagnosesysteme angewendet. Wegen der weitläufigen verteilten Systemarchitektur von Telekommunikationsnetzen wird eine Standardisierung eines Telekommunikations Management Netzwerkes (Telecommunication Management Network TMN) angestrebt [42, 162].

2.2 Expertensysteme

Unter dem Begriff "Künstliche Intelligenz (KI)" (englisch: Artificial Intelligence, AI) werden Forschungsaktivitäten mit dem Ziel, menschliches Verhalten auf Computern nachzubilden, zusammengefaßt. Zum überwiegenden Teil haben die Arbeiten im Gebiet der "Künstlichen Intelligenz" immer noch reinen Forschungscharakter. Lediglich aus dem KI-Teilbereich der wissensbasierten Systeme (Expertensysteme) existieren kommerzielle Anwendungen. Dabei handelt es sich z. B. um wissensbasierte Systeme für komplexe Planungs- und Konfigurationsaufgaben [6] und verschiedene Diagnoseanwendungen [21, 29, 179, 194]. Weitere kommerzielle Anwendungen wissensbasierter Systeme sind zu erwarten.

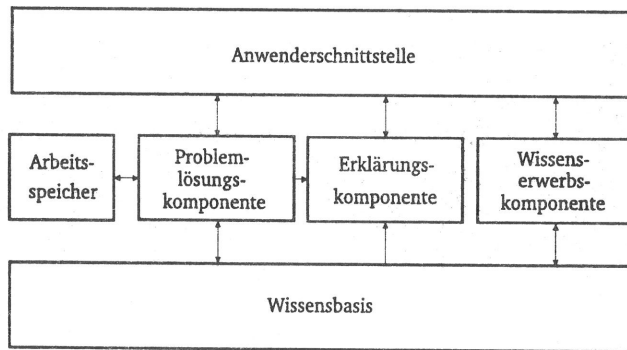


Bild 2.2 Prinzipielle Architektur von Expertensystemen

Den prinzipiellen Aufbau eines wissensbasierten Systems zeigt Bild 2.2. [96, 133, 149, 150, 151, 165] Das für den Problembereich erforderliche Wissen wird in einer Wissensbeschreibungform [62, 144, 165] in der sogenannten Wissensbasis abgelegt. Anhand des in der Wissensbasis gespeicherten Wissens und aktueller, vom Fehlerfall abhängiger, Bewertungen, sucht die Problemlösungskomponente (Inferenzkomponente) nach der Lösung des aktuellen Problems. Die Trennung des Wissens vom eigentlichen ausführenden Programm und dessen Speicherung in einer Wissensbasis ist ein Hauptmerkmal eines wissensbasierten Systems. Durch die Trennung ist die Änderung des Wissensbasisinhalts zur Laufzeit des Systems möglich.

Über die Anwenderschnittstelle werden Bewertungen vom Anwender erfragt oder Ergebnisse ausgegeben.

Die Unterstützung des Anwenders beim Aufbau der Wissensbasis ist Aufgabe der Wissenserwerbskomponente. Die Erklärungskomponente soll dem Anwender Fragen und gefundene Problemlösungen des Systems erläutern und dokumentieren können. Durch die Erklärungskomponente wird die Akzeptanz des Systems erhöht.

Die Wissensdarstellung (Wissensrepräsentation) kann nach unterschiedlichen Prinzipien erfolgen. Die bekanntesten Methoden sind [30, 57, 58, 62, 144, 149, 150, 165]:

- Semantische Netze
- Objekt-Attribut-Wert-Tripel
- Regeln
- Wissensrahmen (Frames)
- Logische Systeme (z.B. Prädikatenlogik)

Die allgemeinste Form der Wissensdarstellung ist das *semantische Netz* (oder semantische Netzwerk). Ein semantisches Netz ist eine Sammlung von Objekten, die auch als Knoten bezeichnet werden. Die Verbindungen (links) der Knoten untereinander werden ebenfalls mit Namen versehen und stellen die Beziehungen der Objekte untereinander dar. Verbindungen werden oft zur Darstellung einer Relation zwischen Einzelfall (Instanz eines Objektes) und einer Klasse von Objekten verwendet.

Eine andere Form Wissen zu repräsentieren, ist die Darstellung als *Objekt-Attribut-Wert Tripel*. Objekte können physische oder begriffliche Einheiten sein. Durch Attribute werden Eigenschaften eines Objektes dargestellt. Der Wert eines Attributs ist der dritte Wert eines Tripels. Durch ihn wird der Zustand eines Attributs in einer bestimmten Situation dargestellt. (z.B.: Ein Attribut eines Objektes trifft zu oder nicht zu.) Die Wissensrepräsentation in dem Tripel Objekt-Attribut-Wert ist ein Spezialfall der Wissensrepräsentation durch semantische Netze. Zwei einfache Relationen treten an die Stelle von sehr vielen Beziehungen.

Regeln dienen der Darstellung von Beziehungen. Mit Regeln können unbewertete Attribute durch bereits bewertete Attribute bewertet werden. Dementsprechend können in semantischen Netzen Netzteile hinzugefügt oder isoliert werden.

Werden sämtliche, zu einem Objekt gehörende, Information in einer zum Objekt gehörenden Datenstruktur abgelegt, dann spricht man von sogenannten *Wissensrahmen* oder *Frames*. In den einzelnen Teilen (Slots) können Attribute, deren Werte, Zeiger auf weitere Objekte, Regeln, Prozeduren zur Berechnung von Werten usw. gespeichert sein. Frames lassen sowohl die deklarative Beschreibung von Wissen als auch die prozedurale Beschreibung von Wissen zu. Die deklarative Beschreibung von Wissen entspricht der Wissensdarstellung in Form von Objekten, Relationen, Attributen usw. Diese Form der Wissensdarstellung ist oft verständlicher bzw. transparenter und kann aufgrund ihrer Modularität einfacher gewartet werden. Als prozedurale Wissensdarstellung wird die Darstellung von Wissen in Form von Algorithmen bezeichnet. Die prozedurale Wissensdarstellung ist im Einsatz effizienter, aber schwieriger zu erstellen und zu warten.

Die Aussagenlogik selbst ist ein allgemeines *logisches System*. Aussagen können entweder wahr oder falsch sein. Aussagen können durch Aussagenverbindungen (Regeln) untereinander verknüpft werden [145, 148, 154, 169]. Die Prädikatenlogik [7, 17, 41] ist eine Erweiterung der

Aussagenlogik. Grundelemente der Prädikatenlogik sind wiederum Objekte. Aussagen über Objekte werden Prädikate genannt. Objekte und Prädikate bilden Aussageverbindungen, die entweder wahr oder falsch sein können [157]. Die durch Objekt und Prädikat gebildeten Aussagen lassen sich zu komplexeren Aussagen verknüpfen. Die Prädikatenlogik ist eine Wissensrepräsentationsform, die mehr in Japan und Europa verbreitet ist.

Der Wissenserwerb stellt die schwierigste Aufgabe beim Aufbau eines wissensbasierten Systems dar. Selten wird die Wissenserfassung direkt durch den Experten eines Wissensgebietes durchgeführt. Vielmehr befragt ein sogenannter Wissensingenieur (Knowledge Engineer) den Experten und setzt das Ergebnis der Befragung in eine geeignete Wissensrepräsentationsform um.

Wissensbasen umfassen Wissen von abgegrenzten Spezialgebieten (Konfigurationsplanung von Rechnersystemen, Diagnose von Getrieben, Leberkrankheiten usw.) Die Abgeschlossenheit des darzustellenden Wissensgebietes ist bislang eine Grundvoraussetzung für die erfolgreiche Anwendung von wissensbasierten Systemen. Die Größe der Wissensbasis hat Einfluß auf deren Konsistenz und auf das Laufzeitverhalten des Gesamtsystems. Die genannten Bedingungen erschweren den Einsatz wissensbasierter Systeme für die Diagnose großer komplexer Systeme. In Kapitel 6 dieser Arbeit wird eine Architektur eines wissensbasierten Systems für die Diagnose komplexer Systeme vorgestellt.

Für die Problemlösung finden allgemein folgende Inferenzmechanismen Anwendung:

- Modus ponens
- Schließen über vages Wissen
- Resolutionsmethoden

Der *Modus ponens* besagt: Ist A wahr, und es gilt die Regel "Wenn A wahr ist, dann ist auch B wahr", dann kann daraus geschlossen werden, daß auch B wahr ist. Mit dem Modus ponens lassen sich neue Fakten aus Regeln und schon bekanntem Wissen ableiten. Viele Systeme arbeiten ausschließlich mit der Anwendung des Modus ponens. Weitere gültige Schlußfolgerungen lassen sich damit nicht ziehen. So kann z. B. nicht geschlossen werden, daß wenn B falsch ist und oben genannte Regel gilt, auch A falsch ist (letztere Schlußfolgerungstechnik wird auch als Modus tollens bezeichnet).

Expertensysteme müssen in der Lage sein, *Schlüsse aus vagem Wissen* zu ziehen. Dazu wird der Grad der Gewissheit, ob Fakten zutreffen oder nicht, durch Bewertungswerte (auch als Konfidenzfaktor oder engl.: certainty factor bezeichnet) angegeben. Für Bewertungswerte werden beliebige Zahlen-Intervalle angenommen, die dann entsprechend interpretiert werden (z.B. wird im Expertensystem MYCIN [21] das Intervall zwischen -1 (sicher nicht) und 1 (sicher) verwendet). Auch Algorithmen der unscharfen Logik (Fuzzy Logic) [123, 156, 196, 198] finden zur Behandlung von unsicherem Wissens Anwendung.

Um herauszufinden, ob eine neue Tatsache abgeleitet von anderen, bekannten Fakten gültig ist, sind *Resolutionsmethoden* notwendig. Zur Durchführung von Resolutionsmethoden findet häufig die Aussagenlogik, auch in Verbindung mit unscharfer Logik, Anwendung. Ziel von Resolutionsmethoden ist es, Regelausdrücke der allgemeinen Form: "Wenn..., dann..." in einen logischen, einfach bewertbaren Ausdruck umzuformen. Logische Systeme (z.B.: Prädikatenlogik) arbeiten mit Resolutionsmethoden anstelle des Modens ponens.

Neben den Mechanismen zur internen Bestimmung von Fakten werden Mechanismen zur Ablaufsteuerung benötigt. Hierzu zählen:

- Rückwärtsverkettung (Backward Chaining)
- Vorwärtsverkettung (Forward Chaining)
- Tiefensuche (Depth First)
- Breitensuche (Breadth First)

Die genannten Mechanismen bestimmen die Richtung, innerhalb eines Objektraumes, in der nach der Problemlösung weiter gesucht wird. Die *Rückwärtsverkettung* versucht, ausgehend von einer vermuteten Lösung, diese zu bestätigen. Dazu werden die zugeordneten Objekte untersucht, die bestätigend zur Vermutung beitragen.

Anhand von bekannten Fakten und Regeln versucht die *Vorwärtsverkettung*, ein mögliches Ziel für die Problemlösung in der Wissensbasis zu bestimmen.

Die Suche nach detaillierten Problemlösungen hat bei der *Tiefensuche* den Vorrang vor der Erfassung von allgemeinem Wissen über den Zustand eines Systems. Letzteres ist Ziel der *Breitensuche*.

Welche Mechanismen Verwendung finden, ist von der aktuellen Problemstellung abhängig. Die Rückwärtsverkettung ist nur sinnvoll anwendbar, wenn zunächst ein mögliches Ziel bestimmt werden kann (z.B.: Die Suche nach einem optimalen Weg zu einem Ziel in einer Stadt). Die Anwendung der Vorwärtsverkettung bietet sich an, wenn zunächst nur irgendwelche Randwerte bekannt sind und kein Ziel bestimmt werden kann. Probleme im Zusammenhang mit Steuerungsaufgaben werden oft mit Hilfe von vorwärtsverkettenden Inferenzmechanismen gelöst. Um möglichst schnell die Problemlösung zu finden, wenden die meisten Systeme die Tiefensuche an. Diese Vorgehensweise entspricht mehr der eines "Spezialisten", während die Anwendung der Breitensuche mit der Vorgehensweise eines "Generalisten" verglichen werden kann.

Für Diagnoseanwendungen erscheint zunächst die Anwendung der Vorwärtsverkettung als Ablaufmechanismus sinnvoll, da das Ziel, die Diagnose, zunächst nicht bekannt ist. Die Anwendung der Rückwärtsverkettung nach der Bestimmung einer vermuteten Teildiagnose sollte bei Diagnosesystemen berücksichtigt werden. Ebenso sollte abhängig von aktuellen Fakten entschieden werden, ob der Mechanismus Tiefensuche oder Breitensuche durchgeführt wird.

2.3 Expertensysteme für die Netzdiagnose

Der Einsatz wissensbasierter Systeme bietet sich im Bereich des Netzmanagement an [13]. Nahezu alle großen Betreiber von Netzwerken (das sind im wesentlichen die großen Telekommunikationsgesellschaften einzelner Staaten) arbeiten an Expertensystemen für unterschiedliche Aspekte des Netzmanagement. Hierzu zählen: Umfassendes Netzmanagement [28, 112, 160, 179, 186, 195], Configuration Management [137, 153], Performance Management [3], Fault Management [18, 23, 109, 124, 128, 130, 141, 142, 143, 193], Verwaltungs- und Planungswerkzeuge [37, 67] und andere.

Kommunikationsnetze werden immer größer und heterogener. Nicht nur die Anzahl der angeschlossenen Endsysteme oder Vermittlungssysteme nimmt laufend zu, auch die Anzahl der verwendeten Technologien innerhalb eines Systems (Satellitentechnik, Glasfaser, Mikrowellen, Paketvermittlung, Leitungsvermittlung, um nur einige Beispiele zu nennen) und der angebotenen Dienste steigt fortlaufend. Rein kombinatorische, prozedurale Systeme für die Aufgaben des Netzmanagements werden, wegen der erforderlichen hohen Komplexität solcher Systeme, schnell unhandlich. So ist der Aufgabenbereich des Netzmanagements geradezu ein ideales Anwendungsgebiet für wissensbasierte Systeme. Für den Bereich des Fault Management von Telekommunikationsnetzen existieren bisher die meisten Ansätze für den Einsatz von Expertensystemen.

Netzexperten analysieren Fehler und Schwächen eines Kommunikationsnetzwerks, um deren Ursachen zu bestimmen. Sie leiten daraus notwendige Reparatur- und Wartungsarbeiten ab. Ziel des Einsatzes von Expertensystemen ist die Verkürzung der Zeit bis zur Fehlerentdeckung und damit bis zur Fehlerbehebung, die Unterstützung des Netzadministrators bei der Fehlerbehebung und evtl. der automatische, direkte Eingriff in Teile des Netzwerkes zur Fehlerbehebung bzw. Fehlervermeidung. Zudem lassen sich Expertensysteme einfach vervielfältigen. Auch der Einsatz von Expertensystemen für die Ausbildung von Netzadministratoren ist möglich.

Eine Reihe von Prototypen von wissensbasierten Diagnosesystemen für die Wartung von Vermittlungsanlagen wurden in den letzten Jahren vorgestellt. Ein Beispiel hierfür ist das System REACT (Real-Time Expert Analysis and Control Tool) für die Diagnose von Vermittlungssystemen (Typ 1AESS), das durch das System SMART (Switching Maintenance and Analysis Repair Tool) ergänzt wird [125]. In die genannten Systeme wurde Wissen von Experten aus Vermittlungskontrollzentren eingebracht. Das System reagiert auf Fehlermeldungen des Vermittlungssystems und schlägt Problemlösungen vor. Dadurch werden die Experten von Routinearbeit entlastet. Ferner lernt das einfache Bedienpersonal durch Anwendung des Systems hinzu.

Das Echtzeit Expertensystem RTES (Real Time Expert System von Bellcore) ist ein Prototyp für die Analyse der Meldungen von digitalen Vermittlungsanlagen [46, 47]. Abgeleitet werden Vorschläge an das Bedienpersonal für vorbeugende und evtl. fehlerkorrigierende Maßnahmen.

Aus den Entwicklungsarbeiten für das Data Management System (DMSTTM) der Northern Telecom wurde die experimentelle Expertensystemshell DRESS (Diagnostic Repair Expert

System Shell) abgeleitet, die speziell auf die interaktive Diagnose und Wartung von Systemen zugeschnitten ist [152].

Nokia Telecommunications und die Universität Helsinki berichten von der Entwicklung eines Systems mit der Bezeichnung DXScope, einem wissensbasierten System für die Diagnose der digitalen Vermittlungsanlage DX200 [147].

Nippon Electric Company (NEC) berichtet von einem Prototyp Diagnosesystem für das NEAX61 Vermittlungssystem. Das System ist vorwiegend für die Unterstützung der Hardwarefehlersuche ausgerichtet [114].

In Europa werden von verschiedenen Herstellern von Vermittlungsanlagen Prototypen für Expertensysteme für Diagnose und Wartung entwickelt. Hierzu zählen unter anderen die British Telecom mit einem System für die Diagnose von TXE4A Vermittlungsanlagen, die französische PTT mit einem System für das Netzmanagement des Telekommunikationsnetzes [181]. Für das italienische paketvermittelnde Datennetz (itapac) wird von einem Expertensystem für das Netzwerk Management berichtet [52]. Für das digitale Vermittlungssystem der Firma Siemens [Elektronisches Wählsystem Digital, EWSD] existiert ein Prototyp eines wissensbasierten Systems für Diagnose und Wartung [11].

Weitere Systeme werden in [1, 192] beschrieben.

Prototypen von wissensbasierten Systemen für die Diagnose und Wartung von lokalen Netzen und Nebenstellenanlagen wurden bislang von einigen Betreibern großer lokaler Netze aus Ermangelung kommerziell verfügbarer Produkte entwickelt [166]. In jüngster Vergangenheit wurde von Hewlett Packard ein geschlossenes System für die Fehlersuche in lokalen Rechnernetzen (Ethernet und/oder Token Ring) angekündigt mit einer Anwenderunterstützung basierend auf Expertensystem-Technologie (HP4980A).

3. Architektur des Meß-, Monitoring- und Diagnosesystems

Im Rahmen der Arbeit wurde ein System zur Unterstützung des Anwenders beim Betrieb lokaler Rechnernetze entworfen und realisiert. Das System ist in folgende drei Teilsysteme untergliedert:

- Verteiltes Meßsystem
- Monitoringsystem
- Diagnosesystem

Bild 3.1 zeigt die Gliederung des Gesamtsystems.

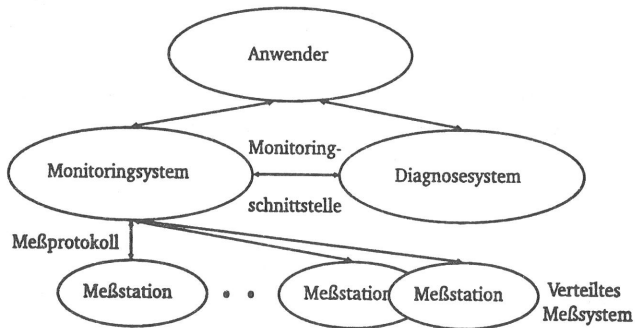


Bild 3.1 Architektur des Meß-, Monitoring- und Diagnosesystems.

Zur Beobachtung des lokalen Rechnernetzes wird ein Meßsystem installiert. Das Meßsystem besteht aus im System verteilten Meßstationen. Die Meßstationen sind modular aufgebaut und können entsprechend der lokalen Anforderungen konfiguriert werden. Die Meßstationen sind die Sensoren des Monitoringsystems. In folgenden Kapitel 4 wird das verteilte Meßsystem näher beschrieben.

Das Meßsystem wird zentral vom Monitoringsystem gesteuert. Die Steuerung erfolgt über ein spezifisches, auf der Schicht 2b nach IEEE 802.2 LLC (Logical Link Control) aufsetzendes Protokoll. Als Übertragungsmedium wird das zu beobachtende, lokale Rechnernetz verwendet. An das Monitoringsystem können über definierte Schnittstellen weitere Meßgeräte angeschlossen werden, sofern sich diese über eine Schnittstelle von der Monitoringstation aus fernsteuern lassen. Das Monitoringsystem verfügt über eine Schnittstelle zum Anwender und zum Diagnosesystem. Der Anwender hat über die Anwenderschnittstelle die Möglichkeit, das verteilte Meßsystem interaktiv zu steuern. Die Architektur des Monitoringsystems wird in Kapitel 5 vorgestellt.

Das Diagnosesystem unterstützt den Anwender bei der Fehlersuche im lokalen Rechnernetz. Die Architektur des Diagnosesystems entspricht der eines Expertensystems mit folgenden anwendungsspezifischen Erweiterungen:

- Monitoringschnittstelle
- Modulare Wissensbasis

Über die Monitoringschnittstelle kann automatisch fehlerspezifisches Wissen erfaßt werden. Dazu werden Meßaufträge an das Monitoringsystem abgesetzt und deren Ergebnisse direkt für die Problemlösung übernommen.

Fehlersuche in komplexen Systemen, wie sie z.B. lokale Rechnernetze darstellen, erfordert eine große Menge an detailliertem Wissen. Hierzu gehört sowohl Wissen über die Architektur und die Besonderheiten des Gesamtsystems als auch über die zum System gehörenden Teilsysteme. Um die Vielzahl von Wissensbereichen geordnet erfassen und beschreiben zu können, wurde der Wissensbasis des Diagnosesystems ein modulare Struktur gegeben.

Eine Wissensbasis besteht aus mindestens einem Wissensbasismodul. In einem Wissensbasismodul wird das Wissen, das für die Diagnose eines Teilsystems benötigt wird, beschrieben. Die Wissensbasismodule sind unabhängig voneinander erstell- und wartbar. Die Architektur des Diagnosesystems wird in Kapitel 6 dargestellt.

4. Ein verteiltes Meßsystem für lokale Rechnernetze nach IEEE 802.3

4.1 Der Standard IEEE 802.3 für lokale Rechnernetze

4.1.1 Geschichte des Standards IEEE 802.3 (Ethernet)

Der Standard IEEE 802.3 / ISO 8802/3 geht zurück auf die Forschungs- und Entwicklungsaktivitäten des Xerox Palo Alto Research Centers. Dort wurde ab 1972 ein lokales Rechnernetz entwickelt, das den Namen Ethernet trägt. Hauptzielsetzungen der Entwicklung waren:

- Entwicklung eines lokalen Netzes mit relativ hohen Datenraten
- Geringe Verzögerungen (keine Speicherung im Netz)
- Einfache Algorithmen
- Unterstützung mehrerer hundert Stationen
- Hohe Zuverlässigkeit, keine zentrale Komponente
- Hohe Stabilität unter Last
- Kostengünstig
- Netzdurchmesser von ca. 1 km

Die ersten Ergebnisse wurden von Xerox 1976 [25] in der Literatur vorgestellt. Die weiteren Entwicklungen wurde dann gemeinsam von den Firmen DEC, Intel und Xerox vorangetrieben. Gegenüber dem experimentalen Ethernet wurde die Datenübertragungsrate von 3 auf 10 Mbit/s angehoben (1980). Die Spezifikation des lokalen Rechnernetzes wurde vervollständigt und präzisiert, um die Kompatibilität heterogener Systeme zu erreichen. Die so entstandene Spezifikation wurde in das Local Network Standard Committee des IEEE eingebracht und, neben anderen Normungsansätzen, im wesentlichen unverändert im ersten zusammenhängenden Statusbericht des IEEE Projekts 802, als Draft B im Oktober 1981 veröffentlicht.

Das IEEE 802-Komitee ist organisiert in sechs Unterkomitees:

802.1	Higher Layer Interface Standard (HILI)
802.2	Logical Link Control (LLC)
802.3	CSMA/CD LAN Standard
802.4	Token Bus Standard
802.5	Token Ring Standard
802.6	Metropolitan Area Network (Distributed Queue Dual Bus, DQDB)

Die Ergebnisse der beiden Untergruppen 802.2 und 802.3 wurden 1985 vom IEEE zum Standard erhoben. Gleichzeitig erhielten sie von der ISO den Status eines Draft International Standard (ISO DIS 8802/2/3)

Mit der Verabschiedung des Standards IEEE 802.3 im Jahre 1985 wurden die Entwicklungen keineswegs abgeschlossen. Neue Optionen und Eigenschaften werden diskutiert, eine Weiterentwicklung des Standards wird angestrebt. Der Standard wurde in jüngster Zeit um die Spezi-

fikation eines Thin-Ethernet (Cheapernet), das die Verwendung eines kostengünstigeren 10 Megabit-Basisband Mediums (10Base2) vorsieht und der Spezifikation einer 10 Megabit Basisband Version für verdrehte Leitungen (10BaseT) mit sternförmiger Netzarchitektur, ergänzt.

4.1.2 Der Standard IEEE 802.3 (Ethernet)

Für das, in dieser Arbeit, vorgestellte Meß-, Monitoring- und Diagnosesystem ist der IEEE 802.3 Standard [25, 70, 82] für lokale Rechnernetze von besonderer Bedeutung. Deshalb sollen in diesem Kapitel die wesentlichen Eigenschaften des Standards erwähnt und gegebenenfalls erläutert werden.

Der Standard IEEE 802.3 definiert den Physical Layer und den Media Access Control Sublayer des Data Link Layers des Schichtenmodells für offene Kommunikationssysteme (Open System Interconnection (OSI) - Schichtenmodell) [36]. Bild 4.1 veranschaulicht die Einordnung des Standards in das OSI-Schichtenmodell.

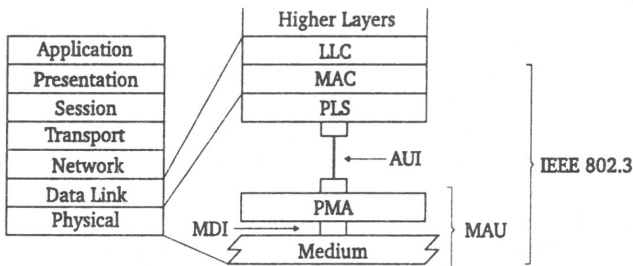


Bild 4.1 Einordnung des Standards IEEE 802.3 in das OSI-7-Schichtenmodell für offene Kommunikation. Durch den Standard wird der Physical Layer und der Media Access Control Sublayer des Data Link Layers abgedeckt. Im Bild werden folgende Abkürzungen verwendet:

- AUI Attachment Unit Interface
- LLC Logical Link Control
- MAC Media Access Control
- MDI Medium Dependent Interface
- PLS Physical Signaling
- PMA Physical Medium Attachment

Durch den Standard werden das Übertragungsmedium, die Art und Weise der Datenübertragung, die maximale Netzausdehnung und der Zugang zum Übertragungsmedium festgelegt.

Ethernet ist ein lokales Rechnernetz in sogenannter Busform. Die einzelnen Stationen werden an das Übertragungsmedium mit einer Medium Attachment Unit (MAU), angeschlossen. Das Übertragungsmedium ist in der Regel ein Koaxialkabel. Seit kurzem wird auch die sternförmige

Verkabelung, realisiert mit verdrehten Telefonleitungen, angeboten. Für das Basisband Ethernet gibt es folgende Kabelarten:

Ethernet, 10Base5, Yellow Cable:

Ein Koaxialkabel, in der Regel gelb, mit einem Außendurchmesser von ca. 10 mm. Das Kabel hat einen Schleifenwiderstand von ca. 8 Ohm/km und einen Wellenwiderstand von 50 Ohm. Die Dämpfung bei 5 MHz beträgt 6 dB, bei 10 MHz 8,8 dB bei jeweils 500 m Kabellänge.

Thin Ethernet, 10Base2, Cheapernet:

Ein Koaxialkabel mit einem Außendurchmesser von ca. 4,6 mm. Das Kabel hat einen Schleifenwiderstand von 50 Ohm/km und einen Wellenwiderstand von 50 Ohm. Die Dämpfung bei 5 MHz beträgt 6 dB, bei 10 MHz 8,8 dB bei jeweils 185 m Kabellänge.

Twisted Pair, 10BaseT:

Verdrehte Zweidrahtleitungen für 10 Mbit/s Ethernet. Als Kabeltypen werden bevorzugt G87 oder IBM Typ 1 in Verbindung mit RJ45- oder IBM Datensteckern eingesetzt. Anstelle der busförmigen Verkabelung tritt bei 10BaseT die sternförmige Verkabelung. Die einzelnen Endgeräte können bis max. 100m (oder 11,5 dB Kabeldämpfung) vom Konzentrator entfernt sein.

Glasfaserkabel, Lichtwellenleiter, Optical Fibre:

Glasfaserkabel haben gegenüber Kupferleitungen den Vorteil der höheren Datensicherheit, der Unempfindlichkeit gegenüber elektromagnetischen Störungen und der flexiblen raumsparenden Verlegung. Der IEEE 802.3 Standard spezifiziert eine maximale Kabellänge von 1 km (62,5/125 Faser) Mit modernen, handelsüblichen Fasern lassen sich jedoch auch Entfernungen bis 2 km realisieren.

Das Zugriffsverfahren, mit dem die einzelnen Stationen auf das Übertragungsmedium zugreifen, wird als Carrier Sense Multiple Access with Collision Detection (CSMA/CD) bezeichnet. Jede, am Netz angeschlossene Station kann zu jedem Zeitpunkt frei auf das Übertragungsmedium zugreifen, um Daten zu senden. Alle über das Übertragungsmedium transportierten Daten können von jeder Station empfangen werden. Diese beiden Eigenschaften werden unter dem Begriff Multiple Access zusammengefaßt.

Da jede Station bei Vorliegen eines Sendewunsches einfach auf das Übertragungsmedium zugreifen kann, können Kollisionen beim Senden auftreten. Um dies zu verhindern, darf jede Station erst dann zu senden beginnen, wenn das Übertragungsmedium frei ist. Dazu hört jede Station auf dem Übertragungsmedium mit. Das Mithören auf dem Übertragungsmedium wird als Carrier Sense bezeichnet. Kollisionen können, bedingt durch das Carrier Sense Verfahren, nur

noch dann auftreten, wenn zwei oder mehrere Stationen gleichzeitig einen Sendewunsch haben und ein freies Übertragungsmedium erkennen.

Dazu muß die Signallaufzeit auf dem Übertragungsmedium, durch die Medium Attachment Units (MAU, Transceiver), Transceiver Kabel, Repeater usw. mitberücksichtigt werden. Eine Station kann an einem Netzpunkt bereits angefangen haben zu senden, während eine Station an einem weit entfernten Netzpunkt immer noch ein freies Übertragungsmedium erkennt. Die maximale Signalübertragungszeit zwischen 2 Netzpunkten wird durch den Standard festgelegt und beträgt:

$$T_p = 25,6 \mu s \quad (\text{Gl. 4.1})$$

Entsteht eine Kollision, dann muß diese von der sendenden Station erkannt werden. Die minimale Übertragungszeit beträgt demnach:

$$2 \cdot T_p = 51,2 \mu s \quad (\text{Gl. 4.2})$$

Die minimale Datenpaketgröße (*minFrameSize*) ergibt sich damit zu:

$$\text{minFrameSize} = \frac{2 \cdot T_p}{\frac{\text{time}}{\text{bit}}} = \frac{51,2 \mu s}{100 \frac{ns}{bit}} = 512 \text{ bit} \quad (\text{Gl. 4.3})$$

Die maximale Datenpaketgröße (*maxFrameSize*) wird durch den Standard auf 1518 Byte festgelegt.

Eine Kollision wird durch die am Übertragungsversuch beteiligten Stationen erkannt und durch zusätzliches Senden von 32 bit (Jamming) verstärkt. Danach wird der Übertragungsversuch abgebrochen. Dieses Verfahren wird als Collision Detection bezeichnet.

Nach einer Kollision warten die betroffenen Stationen eine zufällige Wartezeit T_w :

$$T_w = i \cdot 2 \cdot T_p \quad (\text{Gl. 4.4})$$

bis sie einen neuen Sendeversuch unternehmen. Der Faktor i ist der zufällige Wert einer gleichverteilten Zufallsgröße im Intervall $0 \leq i \leq 2^k$ wobei $k = \min(n, 10)$ ist und n die Durchnumerierung des Wiederholungsversuches für dasselbe Datenpaket. Nach 10 vergeblichen Sendeversuchen wird das Intervall nicht mehr vergrößert. Nach 16 Versuchen wird eine Fehlermeldung erzeugt. Das beschriebene Verfahren wird als "truncated binary exponential back off" Verfahren bezeichnet. Der Algorithmus versucht sich an steigende Netzlasten anzupassen, wobei Neubewerber im Vorteil sind. Da zu den Neubewerbern alle Wiederholungsversuche kommen,

wird deutlich, daß bei hohen Verkehrslasten der Kollisionsoverhead eine zusätzliche, die Kollisionswahrscheinlichkeit erhöhende, Last darstellt.

Nach einem gelungenen Sendeversuch muß eine Station eine Wartezeit (*interFrameGap*) von 9,6 µs einhalten, bis der nächste Sendeversuch unternommen werden darf.

Der Zugang zum Übertragungsmedium wird durch die sogenannten Medium Attachment Units (MAU, Transceiver) bereitgestellt. Die Verbindung zwischen MAU und dem Physical Signaling wird durch das Attachment Unit Interface (AUI) realisiert.

Für die Signalisierung über das AUI sind 4 Signale vorgesehen:

Data In	Datenempfang vom Medium
Data Out	Datensenden auf das Medium
Control In	Signalisierung von der MAU zur Station
Control Out	Signalisierung von der Station zur MAU (optional)

Über das Signal *Data In* werden die auf dem Medium übertragenen Daten in Manchester Codierung an die Physical Signaling Sublayer des Physical Layers weitergeben. Tritt beim Empfang eine Kollision auf, dann wird das Signal *Control In* aktiv.

Von der Station zu sendende Daten werden vom Physical Signaling Sublayer über das Signal *Data Out* in Manchester Codierung zur MAU übertragen. Der gesendete Datenstrom wird gleichzeitig über das Signal *Data In* empfangen. Das Signal *Control In* wird aktiv, wenn beim Senden eine Kollision auftritt.

Das Signal *Control Out* zur Steuerung des Transceivers ist optional und wird selten verwendet. Transceiver, die dieses Signal berücksichtigen, sind ferngesteuert abschaltbar.

Die Signalübertragung erfolgt differentiell über verdrehte Adernpaare. Die Adernpaare werden in einem abgeschirmten Kabel (AUI-Kabel, Transceiverkabel) gemeinsam geführt. Über das Transceiverkabel wird auch die Versorgungsspannung (+12V) für den Transceiver übertragen.

Die MAU wird direkt an das Übertragungsmedium angeschlossen. Die Realisierung der MAU wird als Transceiver, der Anschlußpunkt eines Transceivers an das Übertragungsmedium als Transceiver Access Point (TAP) bezeichnet. Die Verbindung zur Station wird über das AUI-Kabel realisiert, welches auch als Transceiverkabel bezeichnet wird. Das Transceiverkabel darf eine maximale Länge von 50 m haben.

Die MAU ist bei Verwendung von 10Base2 oder 10BaseT in der Regel in den Rechner integriert. Der Anschluß erfolgt bei 10Base2 über BNC T-Kupplungen, bei 10BaseT über RJ45 oder IBM Datenstecker.

Ein Netzsegment darf bei Verwendung von 10Base5 maximal 500 m lang sein. An einem solchen Segment dürfen maximal 100 Medium Attachment Units (MAU) angeschlossen werden. Die MAUs dürfen nur im Abstand ganzzahliger Vielfacher von 2,5 m an das Übertragungsmedium angeschlossen werden. In der Regel ist das verwendete Koaxialkabel in Abständen von 2,5 m gekennzeichnet.

An eine MAU können über einen Schnittstellenvervielfacher mehrere Stationen angeschlossen werden.

Ein durch den Kabeltyp 10Base2 realisiertes Netzsegment darf maximal 185 m lang sein. Die Anzahl der angeschlossenen MAUs ist auf 30 beschränkt. Der Mindestabstand zwischen zwei MAUs ist hier auf 0,5 m festgelegt.

Parameter	Wert
Maximale netzweite Signalübertragungszeit	25,6 us
Maximale Signalübertragungszeit auf einem Koaxialkabel-Segment	2,165 us
Maximale Signalverzögerung auf einem Link-Segment	2,570 us
Wartezeit nach einem erfolgreichen Sendeversuch	9,6 us
Maximale Anzahl von Sendeversuchen	16
Maximale Rahmengröße	1518 byte (12144 bit)
Minimale Rahmengröße	64 byte (512 bit)
Maximale Segmentlänge (10Base2)	185 m
Maximale Segmentlänge (10Base5)	500 m
Maximale Anzahl von Stationen im Gesamtnetz	1024
Maximale Anzahl von MAUs (10Base2) pro Segment	30
Maximale Anzahl von MAUs (10Base5) pro Segment	100
Maximaler Signalweg	2 Link-Segmente und 3 Netzsegmente

Tabelle 4.1: Die wichtigsten Parameter des Standards IEEE 802.3

Über Repeater können mehrere Netzsegmente miteinander verbunden werden. Repeater geben empfangene Datenpakete regeneriert weiter. Der Repeater überträgt jeweils die Datenströme, die auf den angeschlossenen Segmenten empfangen werden, auf das jeweils andere Segment. Wird auf der Empfangsseite eine Kollision bemerkt, dann werden solange Zufallsdaten weiterübertragen, bis die Kollision beendet ist. Eine minimale Sendedauer von 9,6 μ s wird gewährleistet. Da Prüfsumme und Paketlänge des übertragenen Datenpaketes nicht stimmen, wird das übertragene Datenpaket mit Sicherheit als defekt erkannt. Tritt auf der Sendeseite eine Kollision auf, dann werden Zufallsdaten sowohl auf die Sendeseite als auch auf die Empfangsseite gesendet. Damit wird auf der Sendeseite die entstandene Kollision verstärkt, auf der Empfangsseite eine

Kollision herbeigeführt. Damit erkennt die sendende Station, daß es zu einer Kollision gekommen ist und ein neuer Sendeversuch unternommen werden muß.

Segmente ohne Stationsanschlüsse werden als Link-Segmente bezeichnet. Sie können durch Koaxialkabel oder durch Glasfaserkabel realisiert werden. Für die Link-Segmente wird eine maximale Signalverzögerung von 2570 ns angegeben.

Im Standard wird der maximale Signalweg bei Verwendung von 10Base5 oder 10Base2 Verkabelung auf drei normale Segmente und zwei Link-Segmente beschränkt. Bei der Verwendung von Glasfaser Link-Segmenten können die beiden möglichen Link-Segmente zu einem Link-Segment mit doppelter Verzögerungszeit (5140 ns) zusammengefaßt werden. Bild 4.2 zeigt ein Beispiel für den maximalen Ausbau eines Netzes.

Die wichtigsten Werte für lokale Netze nach IEEE 802.3 für die Varianten 10Base5 und 10Base2 sind in Tabelle 4.1 zusammengestellt.

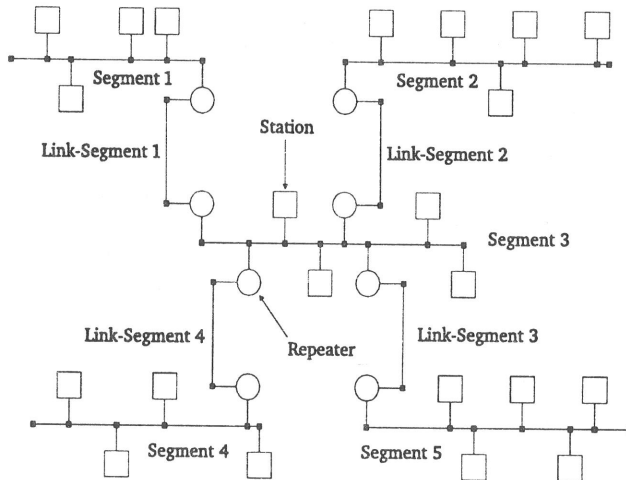


Bild 4.2 Beispiel für den maximalen Ausbau eines lokalen Netzes nach IEEE 802.3 unter Verwendung von 10Base5 oder 10Base2. Der maximale Signalweg zwischen 2 Stationen darf über drei normale Segmente und zwei Link-Segmente führen.

4.1.3 Anwendung des IEEE 802.3 Standards in der Fertigungsautomatisierung

Die Firma Siemens bietet für die Fertigungsautomatisierung das Produkt SINECH1 an, ein lokales Rechnernetz, das auf dem IEEE 802.3 Standard basiert. Verwendet wird als Übertragungsmedium 10Base5, wobei das Kabel eine zusätzliche weitere Abschirmung aufweist. Transceiver werden an das Übertragungsmedium mittels N-Steckern (IEC 68, Teil 1) angeschlossen.

SINEC H1 ist nur eine Netzwerkarchitektur aus dem Hause Siemens für die Fertigungsautomatisierung. Die Firma Siemens schlägt die Anwendung von SINEC H1 für Fertigungszellen vor. Ferner wird von der Firma Siemens angeboten:

- SINEC L2, eine Profibus Realisierung [10, 184] für den Feldbusbereich.
- SINEC L1, ein kostengünstiges Bussystem für die Vernetzung der von Siemens angebotenen Automatisierungsgeräte.
- SINEC H2, ein auf dem Token Bus Standard IEEE 802.4 / ISO 8802/4 basierendes Bussystem für die flächendeckende, werksweite Kommunikation.

SINEC H1 war das erste in Europa breit eingesetzte Bussystem, das die ISO Standards der Schichten 1-4 verwendete. Da Anfang der 80-er Jahre die Standardisierung für die Kommunikation in lokalen Rechnernetzen für die Fertigungsautomatisierung (z.B. Manufacturing Automation Protocol MAP 3.0 [129, 177]) noch nicht abgeschlossen war, wurde von der Firma Siemens eine eigene Kommunikationsarchitektur (SINEC AP) [32, 111] angeboten. Über Gateways kann die Einbindung der SINEC AP Protokollarchitektur in eine MAP-Umgebung realisiert werden [15, 174].

4.2 Architektur des verteilten Meßsystems

Ein lokales Rechnernetz ist ein verteiltes System. Für Überwachung und Diagnose eines lokal verteilten Systems sind im System verteilte Sensoren erforderlich. Die von den Sensoren gelieferten Daten werden zentral erfaßt und ausgewertet [27]. Sie lassen eine Zustandsbeurteilung des Gesamtsystems zu. In einem lokalen Rechnernetz werden als Sensoren Meßstationen installiert, die eine Vielzahl von Messungen zulassen. Die Meßstationen werden zentral vom Monitoringsystem gesteuert. Bild 4.3 zeigt am Beispiel eines einfachen lokalen Rechnernetzes den Einsatz des verteilten Meßsystems.

Unter dem Begriff verteiltes Meßsystem werden alle in einem Rechnernetz installierten Meßstationen zusammengefaßt. Die Meßstationen sind modular aufgebaut. Sie bestehen aus der Basisbaugruppe und verschiedenen Erweiterungsbaugruppen. Bild 4.4 zeigt die modulare Systemarchitektur der Meßstationen.

Die Basisbaugruppe ist eine Rechnerbaugruppe mit Zugang zum lokalen Rechnernetz, 2 seriellen und 3 parallelen Schnittstellen sowie einem Systembus für den Anschluß von Erweiterungsbaugruppen.

Als Erweiterungsbaugruppen stehen zur Verfügung:

- Online Reflektometerbaugruppe für 10Base2 (Cheapernet) und 10Base5 (Ethernet)
- Durchflußmesserbaugruppe für das Attachment Unit Interface (Transceiverkabel) nach IEEE 802.3 Standard
- Speichererweiterungsbaugruppe

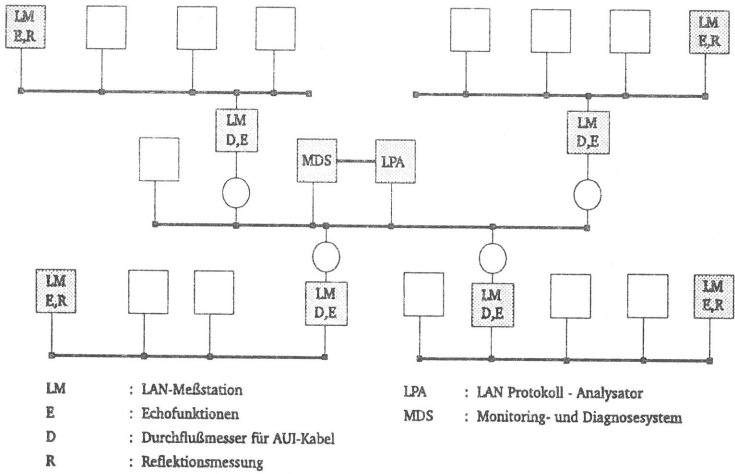


Bild 4.3 Beispiel für den Einsatz des verteilten Meßsystems in einem lokalen Rechnernetz.

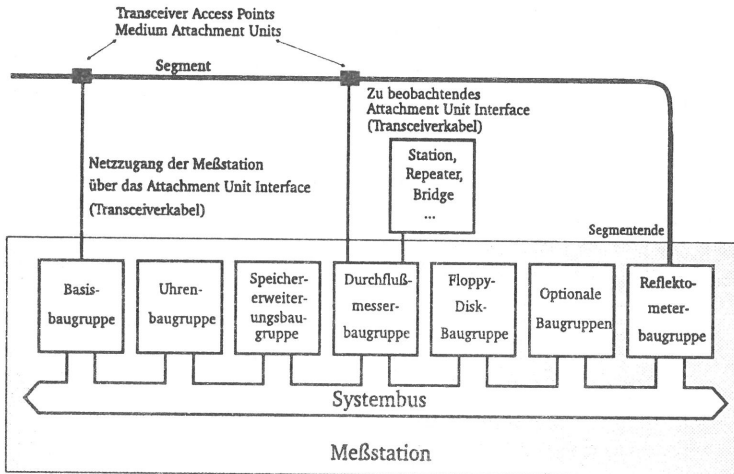


Bild 4.4 Architektur einer Meßstation. Eine Meßstation besteht in ihrer Minimal-konfiguration aus einer Basisbaugruppe. Durch Erweiterungsbaugruppen kann eine Meßstation den lokalen Erfordernissen angepaßt werden.

- Floppy-Disk-Baugruppe
- Uhrenbaugruppe mit batteriegepuffertem Speicher
- u. a.

In den folgenden Unterkapiteln werden die wesentlichen Eigenschaften der Basisbaugruppe und der Erweiterungsbaugruppen erläutert.

4.3 Die Basisbaugruppe

4.3.1 Architektur der Basisbaugruppe

Die Basisbaugruppe ist eine Rechnerbaugruppe, basierend auf dem Intel 80186 Prozessor und dem Koprozessor Intel 82586 [73, 74], für den Zugang zum lokalen Rechnernetz nach IEEE 802.3 Standard. Die Basisbaugruppe verfügt über 2 serielle Schnittstellen (RS232) und drei 8 Bit parallele Schnittstellen. Eine der 3 parallelen Schnittstellen wird zur Ansteuerung von 8 Leuchtdioden zur Statusanzeige verwendet. Für Systemerweiterungen wurde die Basisbaugruppe mit einem Systembus versehen. Der Systembus entspricht im wesentlichen dem Prozessorbussystem des Intel 80186 Prozessors.

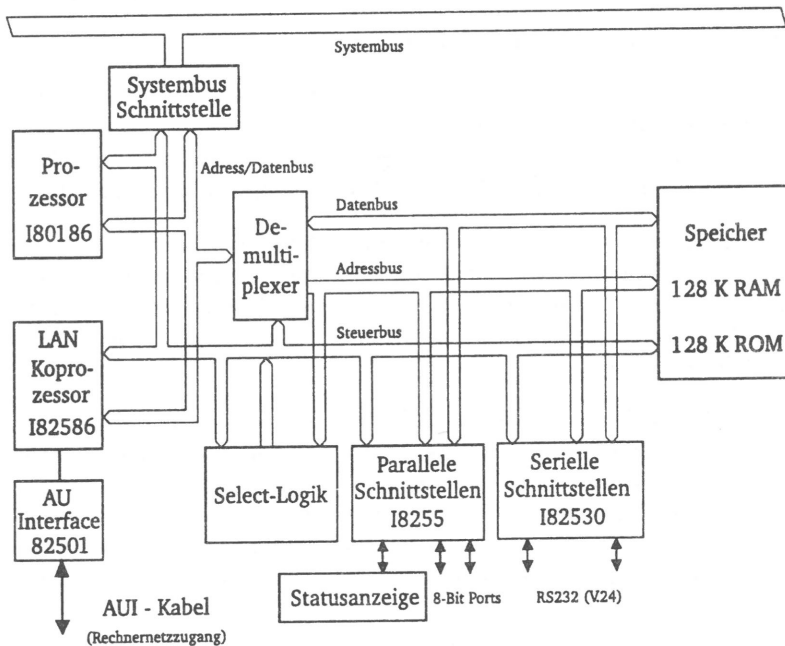


Bild 4.5 Blockschaltbild der Basisbaugruppe

Die Basisbaugruppe [53, 68] kann mit maximal 128 KByte statischem Schreib-/Lesespeicher mit wahlfreiem Zugriff (RAM) und 128 KByte Lesespeicher mit wahlfreiem Zugriff bestückt werden (ROM). Letztgenannter Speicher wird durch elektronisch programmierbare und durch UV-Licht löschbare Speicherbausteine (EPROM) realisiert. Bild 4.5 zeigt ein Blockschaltbild der Basisbaugruppe.

4.3.2 Meßmöglichkeiten der Basisbaugruppe

4.3.2.1 Einteilung der Meßmöglichkeiten

Mit der Basisbaugruppe lassen sich verschiedene Tests und Messungen durchführen [34, 50, 106, 140, 168, 171]. Die möglichen Tests und Messungen können in zwei verschiedene Klassen eingeteilt werden:

- Tests und Messungen im normalen Betrieb:
Die Basisbaugruppe wie eine normale Station am Netz betrieben.

- Tests und Messungen im Monitor-Betrieb (Promiscuous Mode)
Die Basisbaugruppe empfängt alle übertragenen Datenpakete und wertet diese aus.

4.3.2.2 Tests und Messungen im normalen Betrieb

Eine Meßstation verhält sich im normalen Betrieb wie jede Station am Netz. An die Meßstation adressierte Datenpakete werden durch den LAN-Koprozessor in den gemeinsamen Speicherbereich von Prozessor und LAN-Koprozessor geschrieben. Tests und Messungen sind daher in diesem Mode nur durch Protokollfunktionen der höheren Protokollschichten möglich.

Für die Überprüfung, ob Stationen unter bestimmten Protokollen erreichbar sind, sind in den Protokollen sogenannte Echotests vorgesehen. Ausgehend von der Station, die einen solchen Echotest durchführt, wird ein sogenanntes *Echo-Request Paket* an die abzufragende Station abgesendet. Die angesprochene Station antwortet auf eine *Echo-Request Paket* mit einem sogenannten *Echo-Confirm Paket*.

Mit einem Echotest kann überprüft werden, ob eine Station momentan erreicht werden kann. Antwortet eine Station auf ein *Echo-Request Paket* mit einem *Echo-Confirm Paket*, so ist dies ein hinreichender Hinweis darauf, daß die Schicht 1 des lokalen Rechnernetzes zwischen beiden Stationen für die Datenübertragung zur Verfügung steht. Kann ein Echotest mit einer Station nicht erfolgreich durchgeführt werden, dann ist dies noch kein hinreichender Hinweis darauf, daß ein Defekt der Schicht 1 des lokalen Rechnernetzes vorliegt, denn die Station könnte ja, z.B. nicht eingeschaltet sein.

Das auf der Basisbaugruppe implementierte Kommunikationssystem antwortet mit einem *Echo-Confirm Paket* nach dem Standard IEEE 802.2, Logical Link Control (LLC), wenn ein *Echo-Request Paket* desselben Standards empfangen wurde. Vorgesehen wurde jedoch auch die

Durchführung von Echotests verschiedener Protokollarchitekturen, wobei die Basisbaugruppe die Funktion des Testinitiators übernimmt. Implementiert sind Echofunktionen der folgenden Protokollarchitekturen:

- Logical Link Control (LLC) nach IEEE 802.3
- Intel Network Architecture (INA)
- DECnet

Denkbar wäre ebenfalls eine Erweiterung der Echotest Möglichkeiten um die "Ping-Funktion" der Transmission Control Protocol / Internet Protocol (TCP/IP) Protokoll-Architektur.

4.3.2.3 Tests und Messungen im Monitor-Betrieb

Im Monitor-Betrieb der Basisbaugruppe werden alle Pakete, die auf dem lokalen Netz übertragen werden, empfangen. Daher ist eingeschränkt eine Analyse des gesamten Datenverkehrs möglich. Die Analysemöglichkeiten werden durch den verfügbaren Arbeitsspeicher und die Arbeitsgeschwindigkeit des verwendeten Prozessors begrenzt. Für komplexe Filterungen des Datenverkehrs ist die Leistung der Basisbaugruppe zu gering.

Folgende Messungen sind im Monitor-Betrieb der Basisbaugruppe möglich:

Messungen ohne Adressfilterung:

- Zählen aller auf dem Rechnernetz übertragenen Datenpakete
- Zählen aller Datenpakete mit Fehlern, wobei unterschieden wird:
 - Fehler in der Prüfsumme (CRC-Fehler)
 - Zu langes Datenpaket
 - Zu kurzes Datenpaket
- Zählen aller Kollisionen auf dem Netz

Messungen mit Adressfilterung

- Zählen aller Datenpakete von einer bestimmten Station.
- Zählen aller Datenpakete zu einer bestimmten Station.
- Zählen aller Datenpakete von einer bestimmten Station zu einer bestimmten Station.
- Stationspezifisches Zählen der Datenpakete mit Fehlern, wobei die oben genannte Fehler gezählt werden können.
- Erstellen einer Verkehrsmatrix über den Datenverkehr zwischen maximal 9 Stationen. Der restliche Datenverkehr auf dem Netz wird ebenfalls mitangegeben.

4.3.3 Einsatz der Basisbaugruppe und Meßdatenauswertung

4.3.3.1 Allgemeines zum Einsatz der Basisbaugruppe

Da die Basisbaugruppe Grundelement jeder Meßstation ist, stehen auch deren Meß- und Testfunktionen in jeder Meßstation zur Verfügung. Empfehlenswert ist der Einsatz einer Speichererweiterungsbaugruppe, wenn mit der Meßstation Verkehrsmatrizen erstellt oder umfangreiche Datenpaketzählungen mit Filterung durchgeführt werden.

4.3.3.2 Echotest

Am häufigsten verwendet wird die eigene Echofunktionalität des Kommunikationssystems der Basisbaugruppe. In der Regel befindet sich an jedem Segment des lokalen Rechnernetzes mindestens eine Meßstation. Um die Verfügbarkeit des lokalen Rechnernetzes zu überprüfen, wird periodisch vom Monitoringsystem ein Echotest mit den Meßstationen durchgeführt. Bei erfolgreichem Echotest mit einer Meßstation ist sichergestellt, daß die zum Übertragungsweg gehörenden Netzsegmente physikalisch in Ordnung sind. Das Monitoringsystem wird auf einer Station mit Zugang zum lokalen Rechnernetz betrieben. Diese Station wird auch als Monitoring- und Diagnosestation bezeichnet.

In der Regel werden nicht alle Stationen eines lokalen Rechnernetzes mit demselben Kommunikationssystem betrieben. Vielfach werden in den höheren Schichten unterschiedliche, zum Teil auch herstellerabhängige Protokollarchitekturen verwendet. Die herstellerübergreifende Kommunikation wird in der Regel durch Protokoll-Gateways bereitgestellt. Für den Test, ob die Kommunikation zwischen Stationen unterschiedlicher Kommunikationssysteme möglich ist, stehen Echotests verschiedener Protokollarchitekturen zur Verfügung. In einer Meßstation werden zu diesem Zweck die Stationsadressen und die Echotestprotokolle zu testender Stationen in einer Tabelle gespeichert. Angestoßen durch das Monitoringsystem führt die Meßstation Echotests mit anderen Stationen durch. Dabei wird das Protokoll verwendet, das diesen Stationen zugeordnet wurde. Das Ergebnis dieser Echotests wird dann dem Monitoringsystem mitgeteilt.

Durch die Echofunktionen kann überprüft werden:

- Verfügbarkeit der Netzsegmente
- Verfügbarkeit einzelner Stationen

4.3.3.3 Zähl- und Filterfunktionen

Mit den einfachen Zähl- und Filterfunktionen der Basisbaugruppe kann die Auslastung des lokalen Rechnernetzes allgemein und bezogen auf wenige Stationen bestimmt werden. In einem lokalen Rechnernetz nach IEEE 802.3, dessen Segmente über Repeater gekoppelt werden, kann überall dasselbe Verkehrsaufkommen gemessen werden. Für die Durchführung komplexerer Messungen können daher in diesem Fall auch mehrere Stationen gemeinsam verwendet werden.

Die Netzlast lokaler Rechnernetze ist eine zeitabhängige Funktion und weist charakteristische Merkmale auf. Dies gilt besonders für lokale Rechnernetze, die in der Fertigungsautomatisierung eingesetzt werden. Hier schwankt die Netzauslastung weit weniger als in einem Netz, das in einer Büro- oder Entwicklungsumgebung eingesetzt wird [51].

Mit der Basisbaugruppe können folgende Netzparameter gemessen werden:

- Anzahl der Kollisionen / Zeiteinheit
- Anzahl der Pakete / Zeiteinheit
- Anzahl der Bytes / Zeiteinheit
- Anzahl der Fehler (CRC, Alignment) / Zeiteinheit

Diese Netzparameter werden bei normalem Netzbetrieb fortlaufend bestimmt und mit bekannten Werten verglichen. Liegen diese Parameter innerhalb eines Toleranzbereiches, werden keine weiteren Aktionen ausgelöst. Wird der Toleranzbereich verlassen, so wird durch detailliertere Messungen versucht, die Ursache zu bestimmen.

Die Anzahl der Kollisionen sollte bei bekannter Netzlast einen bekannten, durch frühere Messungen bestimmten Maximalwert nicht übersteigen. Wird dennoch eine Zunahme bemerkt, so kann dies folgende Ursachen haben:

- Hardwarefehler in einem Segment
- Fehler in einer Media Attachment Unit (MAU)

Durch geeignete Echotests und Messungen mit Adressfilterung kann versucht werden, die Fehlerursache zu isolieren.

4.4 Die Durchflußmesserbaugruppe

4.4.1 Architektur der Durchflußmesserbaugruppe

Die Durchflußmesserbaugruppe ist eine Erweiterungsbaugruppe für Meßstationen [33, 119]. Für den Betrieb der Durchflußmesserbaugruppe ist eine Basisbaugruppe erforderlich. Mit dem Durchflußmesser kann der Datenverkehr richtungsabhängig gemessen werden, der über das Attachment Unit Interface (AUI) abgewickelt wird. Das AUI wurde in Abschnitt 4.1.2. näher erläutert. Das in der Definition des AUI optional angegebene Signal *Control Out* wird durch die Durchflußmesserbaugruppe nicht berücksichtigt. Bild 4.6 zeigt die Anwendung der Durchflußmesserbaugruppe.

Bild 4.7 zeigt das Blockschaltbild des Durchflußmessers. Die Durchflußmesserbaugruppe steuert die einzelnen Messungen selbstständig. Für die Messungen stehen 4 unabhängige 16 bit Zähler zur Verfügung. Gezählt werden:

- Zur MAU gesendete Pakete (PS)
- Von der MAU empfangene Pakete (PE)

- Kollisionen beim Senden von Paketen (KS)
- Kollisionen beim Empfang von Paketen (KE)

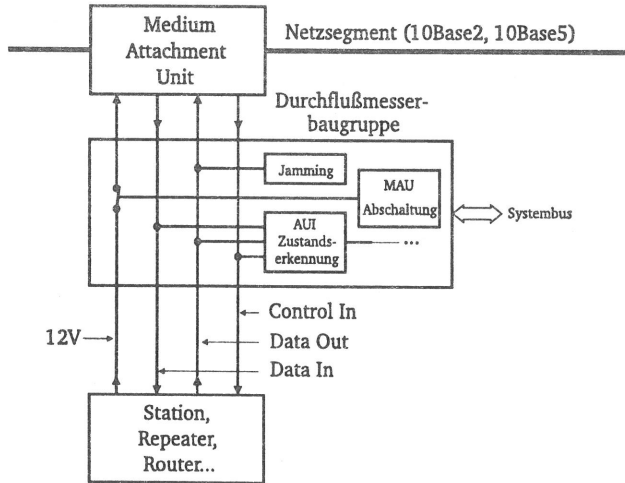


Bild 4.6 Anwendung der Durchflußmesserbaugruppe

Die Aktivitätserkennung generiert aus den differentiellen Leitungssignalen des AUI Interfaces die logischen Signale *CI*, *DI* und *DO*, aus denen die AUI Zustandserkennung die Zählimpulse für die entsprechenden Zähler generiert. Die Signale *CI*, *DI*, und *DO* sind dann aktiv (logisch High), wenn auf den entsprechenden differentiellen Leitungspaaren *Control In*, *Data In* und *Data Out* Aktivität herrscht.

Die AUI-Zustandserkennung wurde als synchrones Steuerwerk realisiert. Das Zustandsübergangsdiagramm der AUI-Zustandserkennung zeigt Bild 4.8.

Ausgehend vom "Initialisierungszustand" *S1* wird anhand der Signale *DI*, *DO* und *CI* unterschieden, ob gerade Daten gesendet oder empfangen werden und ob dabei eine Kollision auftritt. In den "Initialisierungszustand" *S1* geht die AUI-Zustandserkennung immer wenn das Signal *Reset* aktiv ist. Das Signal *Reset* ist ein Signal des Systembusses und wird von der Basisbaugruppe generiert.

Werden die Signale *DI*, *DO* aktiv und bleibt *CI* inaktiv, dann wird gerade kollisionsfrei gesendet. Die AUI-Zustandserkennung geht in den Zustand *S2* "Senden" über. Ist *DI* aktiv, *DO* und *CI* sind inaktiv, dann wird kollisionsfrei empfangen. Die AUI-Zustandserkennung geht in den Zustand *S5* "Empfangen" über.

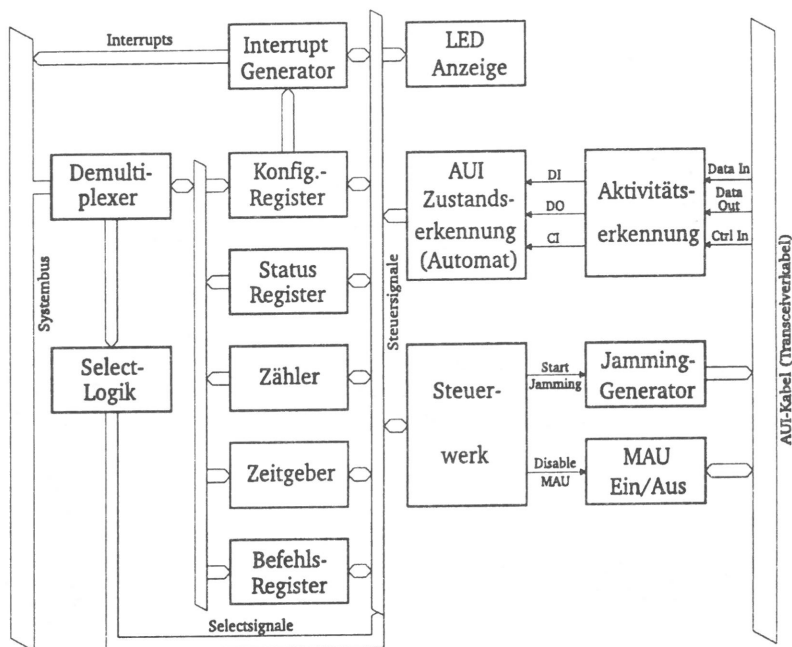


Bild 4.7 Blockschaltbild der Durchflußmesserbaugruppe

Wird im "Initialisierungszustand" S1 das Signal CI aktiv und die Signale DI und DO bleiben inaktiv, dann wird von der MAU der erfolgreiche Abschluß einer Datenübertragung signalisiert. Die AUI-Zustands-erkennung geht in den Zustand S8 "SQE-Test" über.

Werden im "Initialisierungszustand" S1 die Signale CI, DI, DO gleichzeitig aktiv, so liegt eine Kollision beim Senden vor. Die AUI-Zustands-erkennung geht in den Zustand S4 "Kollisionen beim Senden". In S4 wird der Zähler für die Anzahl der Kollisionen beim Senden erhöht.

Entsprechend geht die AUI-Zustands-erkennung in den Zustand S6 "Kollision beim Empfangen", wenn CI und DI aktiv sind und DO inaktiv ist. In S6 wird der Zähler für die Anzahl der Kollisionen beim Empfangen erhöht.

Aus dem Zustand S2 "Senden" wird, wenn die Signale DO und DI inaktiv sind, unter der Voraussetzung, daß das Signal CI inaktiv geblieben ist, in den Zustand S3 "Senden beenden" gegangen. Der Zähler für die gesendeten Pakete wird erhöht. Danach wird wieder in den "Initialisierungszustand" S1 gegangen. Wird im Zustand S2 das Signal CI aktiv, dann ist beim Senden eine Kollision aufgetreten. Die AUI-Zustands-erkennung geht in den Zustand S4 "Kollision beim Senden".

Aus dem Zustand S4 "Kollision beim Senden" wird in den "Initialisierungszustand" S1 zurückgegangen, wenn die drei Signale CI, DI, CO inaktiv geworden sind.

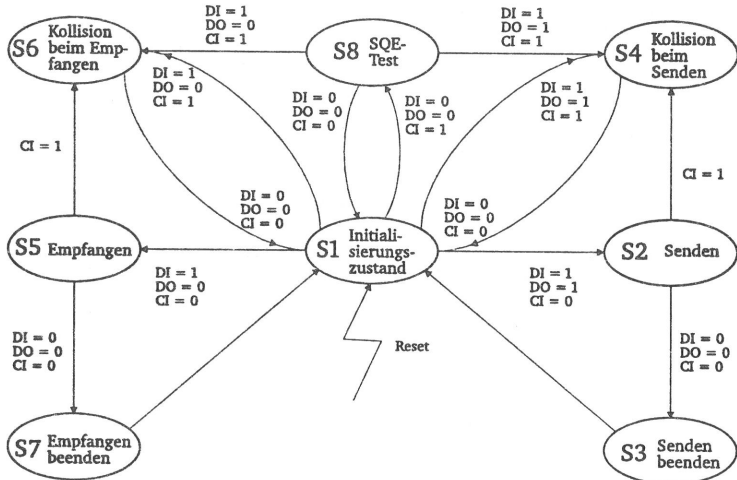


Bild 4.8 Zustandsdiagramm der AUI-Zustandserkennung. An den Zustandsübergängen werden die Bedingungen für einen Zustandsübergang angegeben. Für alle nicht angegebenen Bedingungen bleibt die AUI-Zustandserkennung im aktuellen Zustand. "1" steht für eine aktive, "0" für eine inaktive Eingangsgröße. Aus allen Zuständen wird für ein aktives *Reset* Signal in den Zustand S1 gegangen.

Aus dem Zustand S5 "Empfangen" wird, wenn das Signal DI inaktiv geworden ist, in den Zustand S7 "Empfangen beenden" gegangen. Der Zähler für die Anzahl der empfangenen Pakete wird erhöht. Danach wird in den "Initialisierungszustand" S1 übergegangen. Wird in S5 das Signal CI aktiv, so ist eine Kollision beim Empfangen eingetreten. Die AUI-Zustandserkennung geht in Zustand S6 "Kollision beim Empfangen".

Wird in Zustand S8 "SQE-Test" das Signal DI oder die beiden Signale DO und DI aktiv, wird in die entsprechenden Kollisionserkennungen verzweigt.

Die Durchflußmesserbaugruppe besitzt einen eigenen Zeitgeber (Timer). Die Meßdauer kann in Zweier-Potenzen zwischen 6,6 ms und 40 Tagen und 16 Stunden eingestellt werden.

Wahlweise ist die Erzeugung von Programmunterbrechungen durch den Interruptgenerator bei Ablauf des Zeitgebers oder bei Überlauf eines Zählers möglich.

Die zentrale Steuerung der Durchflußmesserbaugruppe wird durch das Modul Steuerung übernommen. Wesentliche Aufgaben des Moduls sind:

- Starten und Stoppen von Messungen
- Freigabe und Sperren des Zeitgebers
- Starten und Stoppen des Jamming-Generators
- Ein- und Ausschalten der beobachteten MAU

Die Steuerung der Durchflußmesserbaugruppe erfolgt über den Systembus mit dem Konfigurationsregister und dem Befehlsregister. Information über den aktuellen Zustand der Durchflußmesserbaugruppe kann über das Statusregister ausgelesen werden.

4.4.2 Meßmöglichkeiten der Durchflußmesserbaugruppe

4.4.2.1 Messungen an Stationsanschlüssen

Mit der Durchflußmesserbaugruppe kann jedes AUI-Kabel (Transceiverkabel) beobachtet werden [33, 119, 140, 168]. Wird die Durchflußmesserbaugruppe in das AUI-Kabel eines Stationsanschlusses eingeschleift, können folgende Netzparameter gemessen werden:

- Anzahl der von der Station zur MAU übertragenen Datenpakete (PS)
- Anzahl der Kollisionen, die beim Senden auftraten (KS)
- Anzahl der vom Übertragungsmedium empfangenen Datenpakete (PE)
- Anzahl der vom Übertragungsmedium empfangenen Datenpakete, bei denen eine Kollision auftrat (KE)

Der Einsatz der Durchflußmesserbaugruppe an einem AUI-Kabel zum Anschluß einer Station an das lokale Rechnernetz liefert Information darüber, wie groß der Datenverkehrsbeitrag (P_{STAT}) der betreffenden Station zum gesamten Datenverkehrsaufkommen (P_{GES}) am entsprechenden Netzsegment ist.

$$P_{STAT} = PS + KS \quad (\text{Gl. 4.5})$$

$$P_{GES} = PS + PE + KS + KE \quad (\text{Gl. 4.6})$$

Auch kann eine Aussage darüber gemacht werden, ob die Station beim Senden häufig Kollisionen (KS) verursacht und wie hoch der absolute Kollisionsbetrag (K_{ABS}) am zugehörigen Netzsegment ist.

$$K_{ABS} = KS + KE \quad (\text{Gl. 4.7})$$

4.4.2.2 Messungen an Segmentkopplungen

Die Verbindung mehrerer Netzsegmente erfolgt mit Repeatern (Siehe 4.1.2). In AUI-Kabel für Repeater eingeschleifte Durchflußmesserbaugruppen lassen die Bestimmung der Datenverkehrsbeiträge der einzelnen Netzsegmente zum gesamten Datenverkehrsaufkommen zu. Dazu werden Durchflußmesserbaugruppen, wie in Bild 4.3 gezeigt, in das Attachment Unit Interface der Repeater eingeschleift. Die Durchflußmesserbaugruppe sollte immer in das AUI-eingeschleift werden, das zu einer MAU am zu beobachtenden Segment gehört. Soll die Funktion der MAU Abschaltung zur zeitweiligen Abtrennung von Segmenten verwendet werden, dann muß sicher gestellt sein, daß die Meßstation, zu der die Durchflußmesserbaugruppe gehört, nach erfolgter Segmentabschaltung noch über das lokale Rechnernetz erreichbar ist.

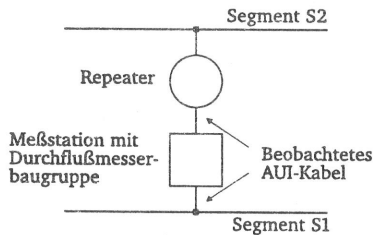


Bild 4.9 Anwendung einer Durchflußmesserbaugruppe bei einer Segmentkopplung über einen Repeater.

Bild 4.9 zeigt eine Anordnung, bei der eine Durchflußmesserbaugruppe in ein AUI-Kabel zum Anschluß eines Repeaters eingeschleift wurde. Mit der im Bild gezeigten Anordnung kann der Verkehrsbeitrag des Segmentes S1 zum Verkehrsaufkommen auf Segment S2 bestimmt werden. Im folgenden werden die einzelnen Übertragungsfälle aufgeführt und die Zählweise der Durchflußmesserbaugruppe näher erläutert.

Bild 4.10 a zeigt den Fall, daß eine Station an Segment 1 sendet. Die Durchflußmesserbaugruppe zählt in diesem Fall ein empfangenes Paket ($PE+$; das "+" - Zeichen nach der Parameterbezeichnung soll andeuten, daß die Zählvariable inkrementiert wird).

Sendet eine Station auf Segment S1, und es tritt auf Segment S1 eine Kollision auf (Bild 4.10 b), so wird diese Kollision von der Durchflußmessereinheit als Kollision beim Empfangen gezählt ($KE+$). Der Repeater hat keine Möglichkeit auf Segment S2 eine Kollision zu generieren. Auf Segment 2 wird die Übertragung des gestörten Paketes mit dem Jamming-Signal abgeschlossen. Auf Segment S2 wird also ein fehlerhaftes Paket übertragen.

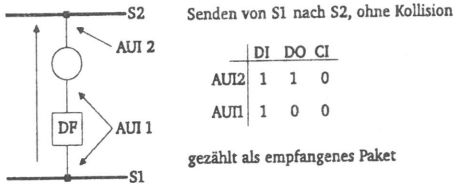


Bild 4.10 a Eine Station an Segment S1 sendet. Das gesendete Datenpaket wird kollisionsfrei übertragen.

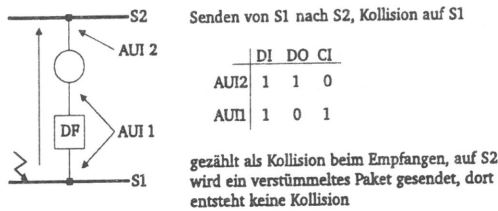


Bild 4.10 b Eine Station an Segment S1 sendet. Auf Segment S1 kommt es zu einer Kollision.

Wenn eine Station an Segment S1 sendet und die Kollision auf Segment S2 eintritt (Bild 4.10 c), dann wird dies vom Durchflußmesser als Kollision beim Empfangen gezählt (*KE+*). Da die Kollision im Sendepfad des Repeaters auftritt, leitet der Repeater eine Kollision auf Segment S1 ein, indem er das Jamming-Signal sowohl auf Segment S1 als auch auf Segment S2 sendet.

Befindet sich die sendende Station an Segment S2 (Bild 4.10 d), dann wird das übertragene Datenpaket als gesendetes Paket gezählt (*PS+*).

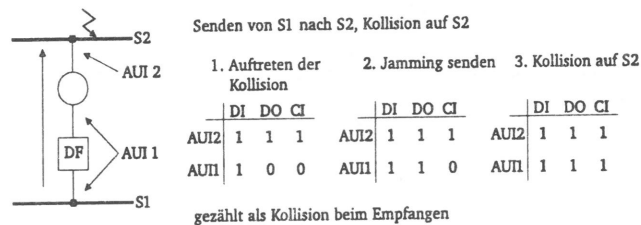


Bild 4.10 c Eine Station an Segment S1 sendet. Auf Segment S2 kommt es zu einer Kollision.

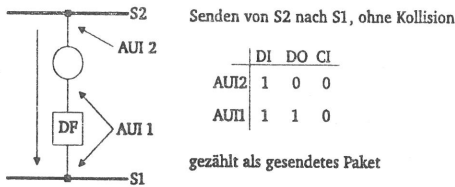


Bild 4.10.d Eine Station an Segment S2 sendet. Das gesendete Datenpaket wird kollisionsfrei übertragen.

Wenn eine Station an Segment S2 sendet und auf dem selben Segment eine Kollision auftritt (Bild 4.10.e), dann wird vom Repeater ein defektes Datenpaket auf Segment S1 übertragen. Dieses wird von der Durchflußmesserbaugruppe als gesendetes Paket gezählt ($PS+$).

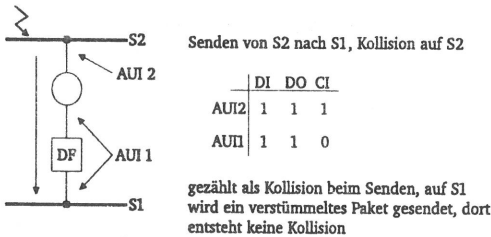


Bild 4.10.e Eine Station an Segment S2 sendet. Auf Segment S2 kommt es zu einer Kollision.

Liegt die Kollisionsursache auf Segment S2, wenn von Segment S1 gesendet wird (Bild 4.10.f), dann wird vom Repeater das Jamming-Signal auch auf Segment S2 gesendet. Dies löst dann auf S2 ebenfalls eine Kollision aus. Von der Durchflußmesserbaugruppe wird dieser Vorgang als Kollision beim Senden gezählt ($KS+$).

Mit Hilfe der Durchflußmesserbaugruppe kann die Anzahl der Kollisionen K_{ABS} entsprechend Gl. 4.7 auf Segment S1 ermittelt werden:

$$K_{ABS} = KS + KE$$

Die Anzahl der empfangenen Datenpakete PE entspricht der Gesamtzahl der Datenpakete, die von den Stationen an Segment S1 gesendet wurden. Hierzu zählen auch die Datenpakete, die von weiteren, evtl. an Segment S1 angeschlossenen, Repeatern in das Segment eingespeist werden.

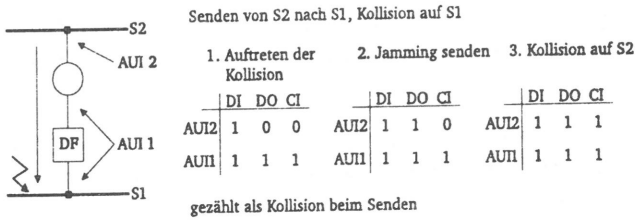


Bild 4.10.f

Eine Station an Segment S2 sendet. Auf Segment S1 kommt es zu einer Kollision.

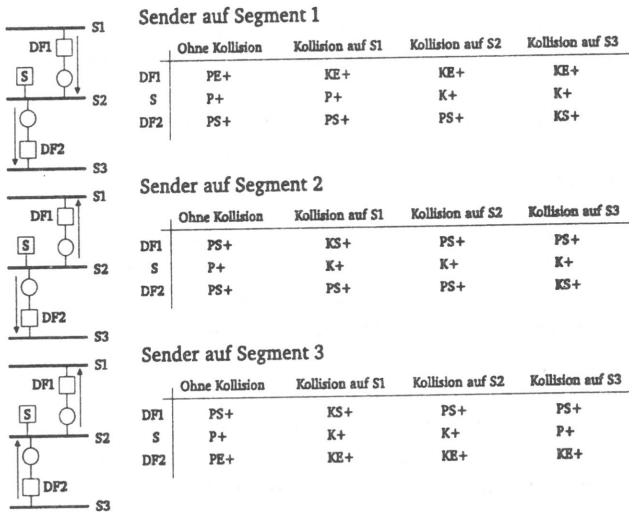


Bild 4.11

Anwendung von 2 Durchflußmesserbaugruppen in einem aus 3 Segmenten bestehenden Netzwerk.

- S Zählerbaugruppe für übertragene Datenpakete und Kollisionen auf S2 (Basisbaugruppe)
- DF1 Durchflußmesserbaugruppe 1
- DF2 Durchflußmesserbaugruppe 2
- PS+ Inkrementieren des Zählers für gesendete Datenpakete
- PE+ Inkrementieren des Zählers für empfangene Datenpakete
- KS+ Inkrementieren des Zählers für Kollisionen beim Senden
- KE+ Inkrementieren des Zählers für Kollisionen beim Empfangen
- P+ Inkrementieren des Zählers für übertragene Datenpakete
- K+ Inkrementieren des Zählers für Kollisionen

Die Anzahl der Datenpakete, die über den Repeater auf S1 gesendet wurden (PS), stellt ein Maß dafür da, wieviele Datenpakete über die beobachtete Repeaterverbindung in Segment S1 eingepeist wurden.

Die Anzahl der Kollisionen, die beim Empfangen (KE) aufgetreten sind, gibt die Anzahl der von Segment 1 ausgehenden Sendeveruche wieder, bei denen es zu einer Kollision irgendwo im lokalen Rechnernetz kam.

Die Anzahl der Kollisionen, die beim Senden auf S1 (KS) auftreten, sind ein Maß dafür, wie oft im Beobachtungspunkt während des Beobachtungszeitraumes ein Sendeveruch durch eine Kollision gestört wurde.

Die gemessenen Werte gelten immer relativ für den einzelnen Beobachtungspunkt. Ein Paket kann, wie in Bild 4.10 e gezeigt, als gesendetes Paket gezählt werden, obwohl es in einem anderen Segment durch eine Kollision gestört wurde. Dies ist am einzelnen Beobachtungspunkt nicht erkennbar. In dem in Bild 4.10 e dargestellten Fall ist es für Segment S1 auch völlig richtig, ein gesendetes Paket zu zählen, da auf Segment S1 auch keine Kollision auftritt. Auf Segment S2 wird jedoch eine Kollision gezählt.

Bild 4.11 zeigt die Anwendung von 2 Durchflußmesserbaugruppen in einem aus 3 Segmenten bestehenden Netzwerk. In den Tabellen wird angegeben, wie in den Durchflußmesserbaugruppen bei den einzelnen Ereignissen gezählt wird. In der mittleren Zeile der Tabellen wird angegeben, wie eine Meßstation S das Verkehrsaufkommen auf Segment S2 zählt.

4.4.3 Meßdatenauswertung

Der Einsatz einer Durchflußmessereinheit an einem Attachment Unit Interface eines Repeaters liefert eine Aussage über das Verhältnis P_{RAT} der am Beobachtungspunkt eingepeisten Anzahl von Datenpaketen PS und den empfangenen Datenpaketen PE .

$$P_{RAT} = \frac{\text{Anzahl der gesendeten Pakete}}{\text{Anzahl der empfangenen Pakete}} = \frac{P_S}{P_E} \quad (\text{Gl. 4.8})$$

In einem lokalen Rechnernetz, das für die Kommunikation einzelner Systeme einer Fertigungszelle eingesetzt wird, unterliegt P_{RAT} geringeren Schwankungen als, z.B. in einem Rechnernetz das für die Büroautomatisierung eingesetzt wird. Dies kann durch die, sich in Intervallen ständig wiederholende, Grundfunktion einer Fertigungszelle begründet werden. P_{RAT} als Funktion der Zeit aufgezeichnet, stellt in diesem Fall oft einen charakteristischen Wert dar. P_{RAT} wird daher bei der Fehlersuche in solchen lokalen Rechnernetzen berücksichtigt.

Wird P_{GES} (Gl. 4.6) an mehreren Stationen oder Repeatern an einem Segment ermittelt und weicht der an einem Meßpunkt gemessene Wert stark von den anderen gemessenen Werten ab

(+/-10%), dann kann vermutet werden, daß ein Defekt der MAU am betreffenden Beobachtungspunkt vorliegt.

4.5 Die Reflektometerbaugruppe

4.5.1 Meßprinzip und Architektur der Reflektometerbaugruppe

Die Reflektometerbaugruppe [8, 14, 53] ist eine Erweiterungsbaugruppe für Meßstationen. Eine Basisbaugruppe ist für den Betrieb der Reflektometerbaugruppe erforderlich. Mit der Reflektometerbaugruppe können bei Netzbetrieb Time Domain Reflektionsmessungen an einem Netzsegment durchgeführt werden. Dazu wird ein Abschlußwiderstand eines Netzsegmentes durch die Reflektometerbaugruppe ersetzt. Bild 4.12 zeigt die Anwendung der Reflektometerbaugruppe an einem Segmentende.

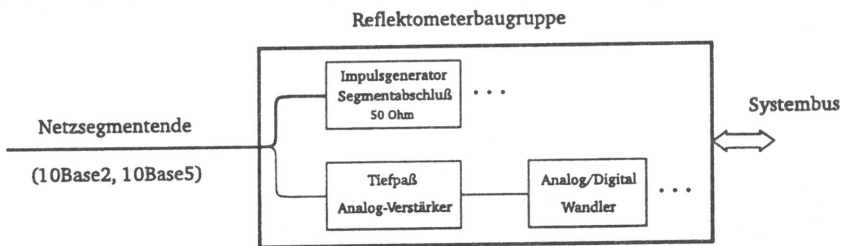


Bild 4.12 Anwendung der Reflektometerbaugruppe

Die Reflektometerbaugruppe digitalisiert den zeitlichen Verlauf der Reflektionsspannung am Segmentende. Man erhält einen für das Segment charakteristischen Datensatz von max. ca. 1 KByte.

Wird am Ende eines Netzsegmentes ein Impuls eingespeist und der nachfolgende Spannungsverlauf über der Zeit aufgezeichnet, so erhält man eine charakteristische Funktion des betreffenden Netzsegmentes. Die so gemessene Funktion wird auch als Reflektionskurve des Netzsegmentes bezeichnet. Anhand der Reflektionskurve kann der Ort von Wellenwiderstandsänderungen entlang des Segmentkabels bestimmt werden. Wellenwiderstandsänderungen entstehen durch den Anschluß von MAUs, falschen Leitungsabschlüssen, Verformungen des Segmentkabels usw.. Die Überprüfung der Reflektionskurve bei Netzbetrieb liefert eine Aussage darüber, ob und gegebenenfalls an welchem Ort, sich die Eigenschaften des Übertragungsmediums verändert haben. Die Auflösung des charakteristischen Datensatzes sollte so hoch sein, daß durch aufeinanderfolgende MAUs verursachte Reflektionen noch unterschieden werden können.

Für ein Netzsegment mit einer maximalen Länge von 500 m berechnet sich die einfache Signallaufzeit t_s aus Segmentlänge s und Signalausbreitungsgeschwindigkeit $v \approx 0,77 c$.

$$t_s = \frac{s}{v} \approx \frac{s}{0,77 \cdot c} \approx \frac{500m}{0,77 \cdot 0,3 \frac{m}{ns}} \approx 2164,5ns \quad (\text{Gl. 4.9})$$

Da die Reflektionskurve am selben Segmentende aufgezeichnet wird, in das der Meßimpuls eingespeist wird, beträgt die eigentliche Meßzeit t_m :

$$t_m = 2 \cdot t_s \approx 4329ns \quad (\text{Gl. 4.10})$$

Die minimale Signallaufzeit t_{MAU} zwischen 2 MAUs an einem Netzsegment nach 10Base5 beträgt:

$$t_{MAU} = \frac{2,5m}{0,77 \cdot c} \approx 10,8ns \quad (\text{Gl. 4.11})$$

Reflektionen von 2 aufeinanderfolgenden MAUs werden am Meßpunkt mit einer zeitlichen Differenz von $2 \cdot t_{MAU}$ aufgezeichnet. Dies liegt daran, daß der Meßimpuls erst nach der Zeit t_{MAU} die zweite MAU erreicht. Die dort hervorgerufene Reflektion entsteht um t_{MAU} später als die Reflektion an der ersten MAU. Bis zum Meßpunkt benötigt die an der zweiten MAU entstandene Reflektion nochmals die Zeit t_{MAU} mehr, da sie die Entfernung zwischen den beiden MAUs zurücklaufen muß.

Um die Reflektionen, die von 2 aufeinanderfolgenden MAUs im Abstand von 2,5 m verursacht werden, unterscheiden zu können, wurde für die Meßimpulsdauer $T_{Meß}$ der Wert 10 ns gewählt. Bild 4.13 erläutert an einem Impulsdiagramm die Entstehung einer Reflektionskurve.

Der am Segmentende eingespeiste Meßimpuls mit der Impulsdauer $T_{Meß}$ erreicht nach der Zeit t_1 den Ort x_1 auf dem Segmentkabel. Am Ort x_1 befindet sich eine MAU. Die durch diese MAU verursachte minimale Wellenwiderstandsänderung ruft eine kleine Reflektion hervor. Zum Zeitpunkt t_1 wird also ein Teil des Meßimpulses reflektiert. Zum Zeitpunkt $t_1' = 2 t_1$ kann diese Reflektion an dem Segmentende, an dem auch der Meßimpuls eingespeist wurde, gemessen werden.

Zum Zeitpunkt t_2 :

$$t_2 = t_1 + t_{MAU} \quad (\text{Gl. 4.12})$$

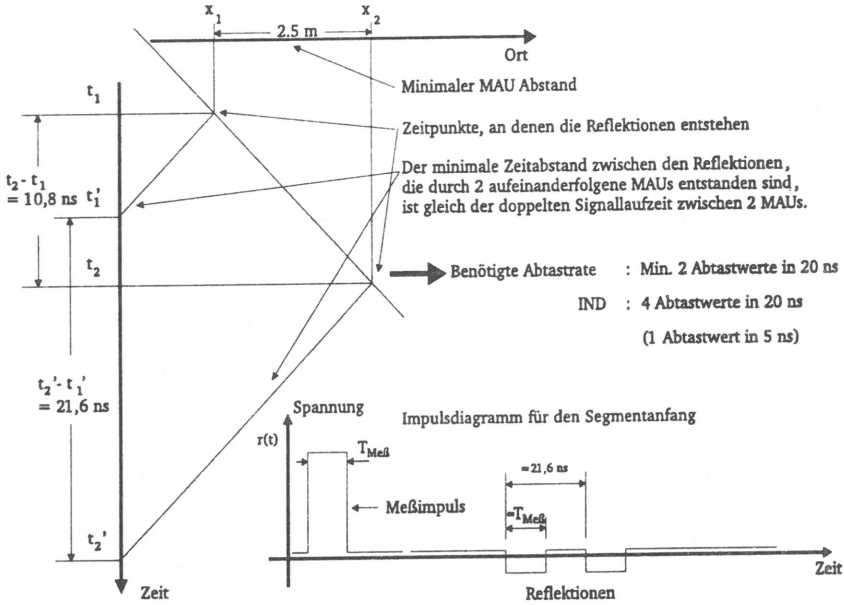


Bild 4.13 Impulsdiagramm zur Reflektionsmessung an einem Netzsegment nach IEEE 802.3

erreicht der Meßimpuls die MAU an Ort x_2 , die in minimal zulässigem Abstand von 2,5 m von der MAU an Ort x_1 am Netzsegment angebracht wurde. Auch hier wird eine kleine Reflexion generiert, die zum Segmentende zurückläuft. Die zurücklaufende Reflexion kann am Segmentende zum Zeitpunkt $t_2' = 2 t_2$ gemessen werden. Die am Segmentende in der Reflektionskurve aufgezeichneten Reflexionen haben den zeitlichen Abstand von t_{diff} :

$$t_{diff} = t_2' - t_1' = 2t_{MAU} \approx 21,6 \text{ ns} \quad (\text{Gl. 4.13})$$

Um die Abtastfrequenz für die geforderte Auflösung zu bestimmen, wird angenommen, daß mehrere MAUs in minimal zulässigem Abstand von 2,5 m an einem Segment angebracht sind. In einfacher Näherung entsteht dann als Reflexionssignal kurzzeitig ein periodisches Signal mit der Grundfrequenz f_{Ref} von:

$$f_{Ref} = \frac{1}{T_{Ref}} \approx 50 \text{ MHz} \quad (\text{Gl. 4.14})$$

Um aus der Reflektionskurve einen charakteristischen Datensatz für das Netzsegment zu gewinnen, muß dem Signalverlauf nach dem Abtasttheorem von Shannon mindestens alle 10 ns ein Abtastwert entnommen werden [75, 183]. Die Reflektometerbaugruppe entnimmt dem Reflektionsverlauf alle 5 ns einen Abtastwert.

Dazu werden 2 Messungen mit äquidistanten Abtastzeitpunkten von 10 ns, mit einem zeitlichen Versatz von 5 ns, durchgeführt. Bild 4.14 erläutert das angewandte Sampling-Verfahren.

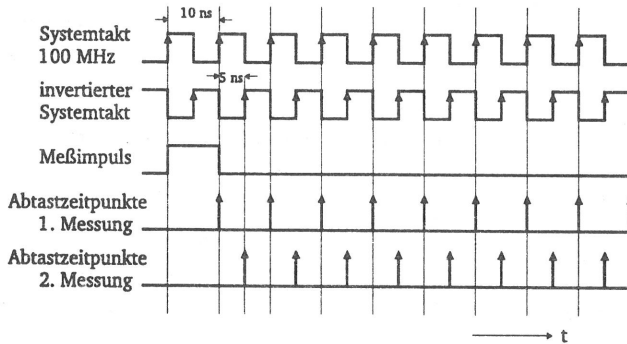


Bild 4.14 Der Meßimpuls wird durch das Steuerwerk aus einer Taktperiode des 100 MHz Systemtaktes abgeleitet und durch die Impulsstufe in das Segmentende eingespeist. Das Meßsystem wird während der ersten Messung mit dem Systemtakt, während der zweiten Messung mit dem invertierten Systemtakt betrieben. Durch Umsortieren der in beiden Messungen erhaltenen Daten erhält man Meßwerte im zeitlichen Abstand von 5 ns. Voraussetzung ist ein Tastverhältnis des Systemtaktes von 1:1.

Bild 4.15 zeigt das Blockschaltbild der Reflektometerbaugruppe. Der Segmentabschluß wird im Analogteil durch den Innenwiderstand des Impulsgenerators gebildet. Der Impulsgenerator speist den Meßimpuls mit einer Amplitude von -2 V in das Netzsegment ein, also genau derselben Amplitude, mit der auch die Datenübertragung durchgeführt wird.

Für die Reflektionssignalerfassung ist ebenfalls ein breitbandiger Meßverstärker am Segmentende angeschlossen. Der Meßverstärker dient gleichzeitig der Signalbegrenzung.

Nach einem Tiefpaß mit einer Grenzfrequenz von $f_g \approx 100$ MHz wird das analoge Signal dem Analog/Digital-Wandler zugeführt. Für die Realisierung wurde ein 8 bit A/D-Wandler mit einer maximalen Abtastfrequenz von 100 Msample/s verwendet. Die Daten werden direkt in einen Speicher geschrieben. Die aktuelle Speicheradresse wird durch einen Zähler bestimmt. Analog/Digital-Wandler, Speicher und Adresszähler sind mit Hilfe von Bausteinen der emittergekoppelten Logik (Emitter Coupled Logic, ECL) realisiert worden.

Das Steuerwerk zur Durchführung der Messungen wurde mit einem programmierbaren ECL-Logik Baustein (Programmable Logic Array PAL) realisiert. Die Anwendung von Bausteinen einer ECL-Bausteinfamilie war notwendig, um mit einer Taktfrequenz von 100 MHz arbeiten zu können. Abhängig von einem Bit, das über das Systembusinterface gesetzt werden kann, wird vom Steuerwerk der invertierte oder der normale Systemtakt für die Abtastung des Reflektions-signales verwendet.

Das Systembus-Interface dient zum Anschluß der Reflektometerbaugruppe an eine Basisbaugruppe. Aufgabe des Systembus-Interfaces sind:

- Pegelanpassung des ECL-Teils an den TTL-Teil
- Zuschalten der Spannungsversorgung des ECL-Teils bei Messungen
- Steuerung des Datenauslesens
- Steuerung der Adressierung der einzelnen Blöcke der Reflektometerbaugruppe

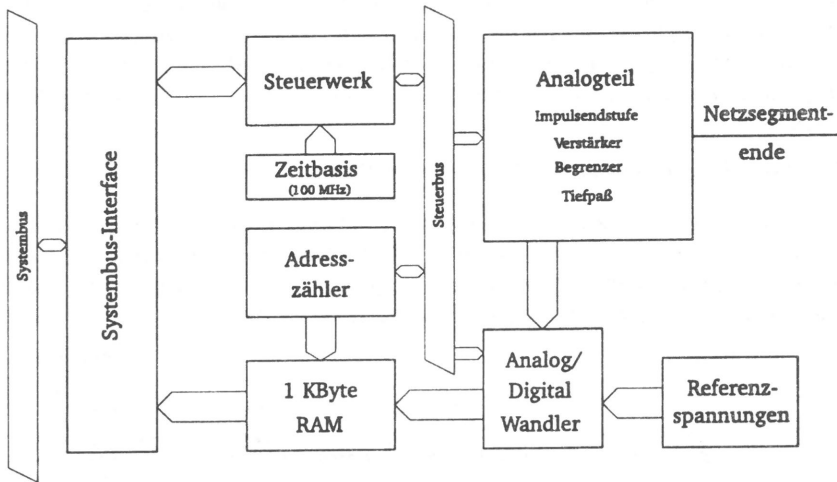


Bild 4.15 Blockschaltbild der Reflektometerbaugruppe

Die Spannungsversorgung des ECL-Teils der Reflektometerbaugruppe wird vor der Durchführung einer Teilreflektionsmessung programmgesteuert eingeschaltet. Danach wird der Typ der durchzuführenden Teilmessung eingestellt (Systemtakt oder invertierter Systemtakt). Nun hört die Basisbaugruppe auf das Netzsegment und wartet, bis auf dem Netzsegment keine Datenübertragung stattfindet. Findet keine Datenübertragung statt, dann wird die Messung gestartet. Eine Teilmessung dauert, wie oben erläutert $t_m \approx 4,4 \mu s$. Während dieser Zeit hört die Basisbaugruppe weiter auf das Medium. Wird eine Datenübertragung erkannt, dann wird die Messung verworfen und ein erneuter Meßversuch gestartet.

Kollidiert der Meßimpuls mit dem Beginn eines Datenpaketes, so wird das Datenpaket dadurch nicht gestört. Die übertragenen Datenpakete beginnen mit einer Präambel von 7 Byte Länge, gefolgt von einem Start-Frame-Delimiter (SFD) mit einer Länge von einem Byte. Dabei wird, beginnend mit einer "1", abwechselnd "0" und "1" übertragen. Dadurch entsteht kurzzeitig ein periodisches Signal mit einer Grundfrequenz von 10 MHz. Lediglich die letzten beiden übertragenen Bits des SFD sind beide "1". Die Präambel und der SFD dienen zur Aufsynchro- nisierung der Empfänger. Präambel und SFD haben die zeitliche Länge T_{PSFD} von:

$$\begin{aligned} T_{PSFD} &= T_{bit} \cdot (L_{Präambel} + L_{SFD}) && \text{(Gl. 4.15)} \\ &= 100 \frac{ns}{bit} \cdot (7 + 1) \cdot 8bit \\ &= 6,4 \mu s \end{aligned}$$

Der Meßimpuls überlagert sich im Fall einer Kollision mit Präambel und SFD. Da der Meßimpuls eine Impulsdauer von 10 ns hat, sind die wesentlichen Frequenzanteile seines Spektrums um den Faktor 10 höher als die Grundfrequenzen von Präambel und SFD. Die Auf- synchronisation der Empfänger am Netzsegment wird deshalb durch den Meßimpuls nicht gestört.

4.5.2 Meßmöglichkeiten der Reflektometerbaugruppe

Mit der Reflektometerbaugruppe ist es möglich, die Reflektionskurve eines Netzsegmentes nach IEEE 802.3 bei Netzbetrieb zu messen. Wie unter 4.5.1 beschrieben, werden dazu 2 Teilmes- sungen durchgeführt. Bild 4.16 zeigt das Layout eines einfachen Netzsegmentes mit 7 Anschluß- punkten von Transceivern und einer insgesamten Netzlänge von 25 m.

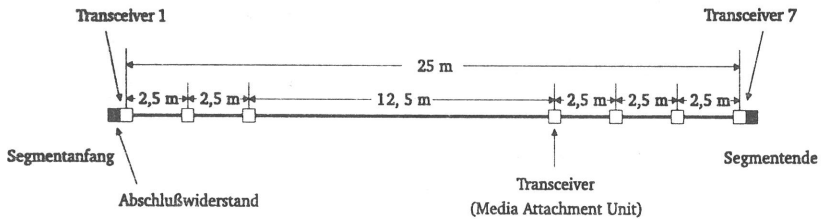


Bild 4.16 Layout eines einfachen Netzsegmentes (SINEC H1 Testsegment am Institut für Nachrichtenvermittlung und Datenverarbeitung)

Bild 4.17 zeigt mit einem Tektronix 1503 Kabeltester analog aufgezeichnete Reflektionskurven. Die Messungen wurden durch Anschluß des Kabeltesters an Transceiver 1 durchgeführt. Dazu wurde der Abschlußwiderstand bei Transceiver 1 entfernt. Mit dem Tektronix Kabeltester kann eine Reflektionskurve nicht bei Netzbetrieb bestimmt werden. Bild 4.18 zeigt mit der Reflektometerbaugruppe gemessene Reflektionskurven. Für die Messungen ersetzte die Reflektometerbaugruppe den Abschlußwiderstand bei Transceiver 1.

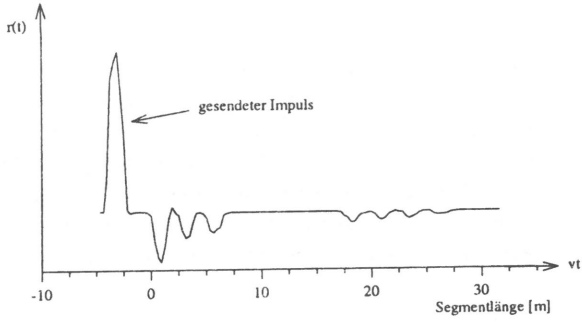


Bild 4.17a. Richtig abgeschlossenes Segmentende

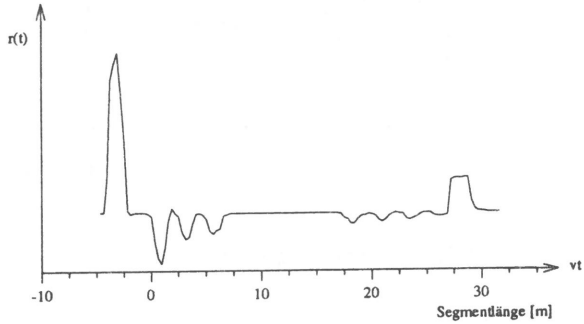


Bild 4.17b. Offenes Segmentende

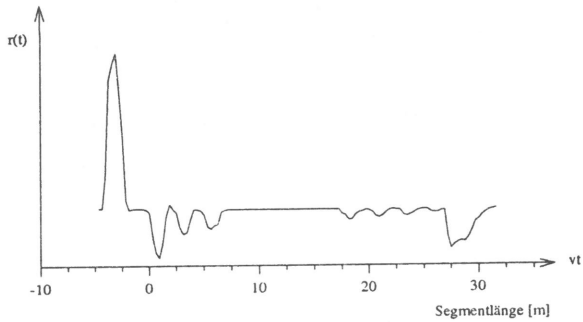


Bild 4.17c. Kurzschluß am Segmentende

Bild 4.17 a-c Analog gemessene Reflektionskurven des in Bild 4.16 dargestellten Netzsegmentes (Tektronix Kabeltester 1503). Es wird ein Meßimpuls mit positiver Amplitude verwendet. Für die Messungen bei offenem bzw. kurzgeschlossenen Segmentende (Bild 4.17 b, c) wurde das Segment durch ein Kabelstück mit ca. 2 m nach Transceiver 7 verlängert. Der Nullpunkt der Abszisse der graphischen Darstellung des Reflektionsverlaufes wurde auf den Beginn der Reflektion gelegt, die durch Transceiver 1 hervorgerufen wird.

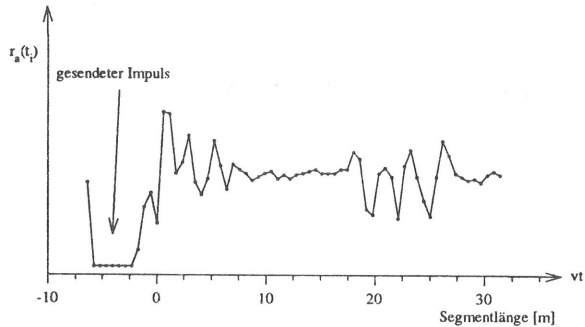


Bild 4.18 a. Richtig abgeschlossenes Segmentende

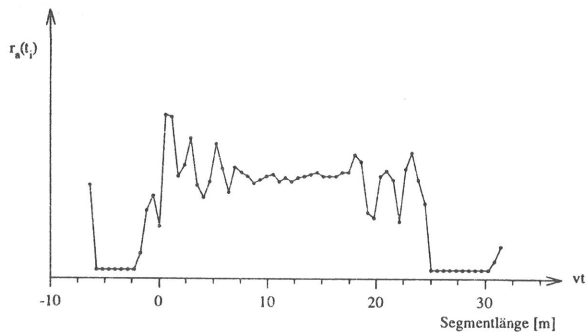


Bild 4.18 b. Offenes Segmentende

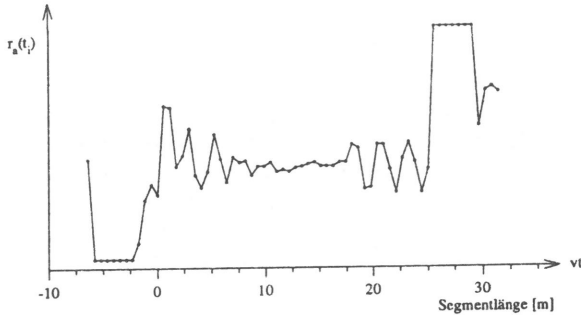


Bild 4.18 c. Kurzschluß am Segmentende

Bild 4.18 a-c Reflektionsverläufe des in Bild 4.16 dargestellten Netzsegmentes, gemessen mit der Reflektometerbaugruppe. Die einzelnen Abtastwerte $r_d(t_i)$ wurden durch Linien miteinander verbunden. Es wird ein Meßimpuls mit negativer Amplitude verwendet. Für die Messungen bei offenem und geschlossenem Segmentende (Bild 4.18 b, c) wurde das Segmentkabel direkt bei Transceiver 7 geöffnet bzw. kurzgeschlossen. Dies ist beim Vergleich mit den Bildern 4.17 b, c zu beachten. Der Nullpunkt der Abszisse wurde auf den Beginn der Reflektion gesetzt, die durch Transceiver 1 hervorgerufen wird.

4.5.3 Meßdatenauswertung

4.5.3.1 Allgemeines zur Meßdatenauswertung

Die Reflektionskurve $r(t)$ wird zu den diskreten Zeitpunkten $i \cdot T_s$ abgetastet. Man erhält dadurch die diskreten Abtastwerte der Reflektionskurve $r_d(i \cdot T_s) = r_i$.

Die Abtastwerte r_i einer Reflektionsmessung werden auch als Musterdatensatz bezeichnet und können als Vektor \underline{r} der Dimension m aufgefaßt werden. Die Musterwerteanzahl m gibt die Anzahl der Werte an, die zu einem Musterdatensatz gehören.

$$\underline{r} = (r_1, r_2, r_3, \dots, r_m) \quad (\text{Gl. 4.16})$$

Die Musterwerteanzahl m eines Musterdatensatzes wird durch die bekannte Segmentlänge s_s und einem zu addierenden Offset O (in der Regel wird $O = 10$ gewählt) bestimmt.

Da alle $T_s = 5 \text{ ns}$ ein Abtastwert dem Reflektionssignal entnommen wird, erhält man eine örtliche Auflösung Δs von:

$$\Delta s = \frac{1}{2} \cdot v \cdot T_s \approx \frac{1}{2} \cdot 0,77 \cdot 0,3 \frac{m}{ns} \cdot 5ns = 57,75cm \quad (\text{Gl. 4.17})$$

Der Faktor $\frac{1}{2}$ rührt daher, daß das Reflektionssignal bis zur Meßstelle immer genau die doppelte Zeit benötigt, wie der Meßimpuls bis zum Erreichen der Reflektionsstelle, da das reflektierte Signal zum Segmentanfang zurücklaufen muß. Die Musteranzahl m ergibt sich zu:

$$m = \frac{\text{Segmentlänge}}{\Delta s} + \text{Offset} = \frac{s_s}{\Delta s} + O \quad (\text{Gl. 4.18})$$

Nach Installation eines Netzsegmentes wird anhand von Reflektionsmessungen ein Sollmusterdatensatz \underline{s} mit der Werteanzahl m_s bestimmt. Für die Erstellung des Sollmusterdatensatzes werden n Messungen durchgeführt. Der Musterwert s_i des Sollmusterdatensatzes wird durch Mittelwertbestimmung aus n Werten bestimmt:

$$s_i = \frac{1}{n} \sum_{k=1}^n r_{ik} \quad (\text{Gl. 4.19})$$

r_{ik} ist der i -te Meßwert der k -ten Reflektionsmessung.

Für die Überprüfung eines Netzsegmentes wird analog der Bestimmung des Sollmusterdatensatzes ein Istmusterdatensatz \underline{i} bestimmt. Beide Datensätze werden nun verglichen. Ergibt der Vergleich Abweichungen, so muß davon ausgegangen werden, daß sich am Netzsegment etwas verändert hat. Der Ort der Abweichung läßt sich auf $\pm \Delta s$ genau bestimmen.

Bei der Ermittlung des Istmusterdatensatzes \underline{i} können Meßfehler auftreten, die beim Vergleich von \underline{i} und \underline{s} berücksichtigt werden müssen [5, 54]. Ein direkter Vergleich der einzelnen Meßwerte ist daher nicht möglich. Die wesentlichen Meßfehler sind:

- **Jitter der Abtastfrequenz**
Der Istmusterdatensatz wird im allgemeinen mit einer leicht unterschiedlichen Abtastfrequenz gegenüber dem Sollmusterdatensatz erstellt.
- **Verstärkungsschwankungen des Eingangsverstärkers**
Die Verstärkung des Eingangsverstärkers kann schwanken. Eine Schwankung der Referenzspannung des Analog/Digital-Wandlers wird ebenfalls als Verstärkungsschwankung bemerkbar.
- **Offset-Spannung**
Im Verstärker des Analogteils der Reflektometerbaugruppe kann eine Offset-Spannung entstehen, welche sich zu den einzelnen Meßwerten addiert.

- **Rauschen**

Die einzelnen Meßwerte werden durch die Überlagerung von Rauschgrößen verfälscht. Das Rauschen wird größtenteils durch die Mittelwertbildung beim Erstellen der Musterdatensätze eliminiert.

Die Eliminierung der Meßfehler aus den Musterdatensätzen wird vor dem eigentlichen Vergleich der Musterdatensätze vorgenommen.

Der Jitter in der Abtastfrequenz bewirkt ein Stauchen oder Dehnen des Signals auf der diskreten Zeitachse. Die Eliminierung dieses Fehlers könnte durch die Rekonstruktion des Reflektions-signales aus den einzelnen Abtastwerten erfolgen. Die diskrete Reflektionskurve $r_d(t)$ wird aus der kontinuierlichen Reflektionskurve $r(t)$ durch Multiplikation mit der Funktion $f_{abt}(t)$:

$$f_{abt}(t) = \sum_{i=-\infty}^{\infty} \delta(t - iT_s) \quad (\text{Gl. 4.20})$$

gewonnen. Die Funktion $f_{abt}(t)$ besteht aus, im Abstand T_s aufeinanderfolgenden, Dirac-Impulsen. Die Musterwerte eines Musterdatensatzes sind die einzelnen Werte der diskreten Reflektionskurve $r_d(t)$ im Zeitintervall $[0..m \cdot T_s]$. Der Funktion $f_{abt}(t)$ entspricht im Frequenzbereich die Funktion $F_{abt}(f)$:

$$F_{abt}(f) = \frac{1}{T_s} \sum_{i=-\infty}^{\infty} \delta(f - if_s) \quad \text{mit} \quad f_s = \frac{1}{T_s} \quad (\text{Gl. 4.21})$$

Die Multiplikation von $r(t)$ mit $f_{abt}(t)$ im Zeitbereich entspricht einer Faltung der Fouriertransformierten dieser beiden Funktionen im Frequenzbereich:

$$r_a(t) = r(t) \cdot f_{abt}(t) \quad \circ \rightarrow \quad R_a(f) = R(f) * F_{abt}(f) \quad (\text{Gl. 4.22})$$

Damit ergibt sich für $R_a(f)$:

$$\begin{aligned} R_a(f) &= R(f) * \frac{1}{T_s} \sum_{i=-\infty}^{\infty} \delta(f - if_s) \\ &= \frac{1}{T_s} \sum_{i=-\infty}^{\infty} R(f - if_s) \end{aligned} \quad (\text{Gl. 4.23})$$

Das Spektrum des Signales $r(t)$ wird alle f_s wiederholt. Das Originalsignal kann durch Filterung rekonstruiert werden, wenn die Spektren sich nicht überlappen. Dazu muß die Abtastfrequenz größer als die doppelte Grenzfrequenz des Signals sein. Die Filterung kann durch Multiplikation im Frequenzbereich mit einer Rechteckfunktion $Q(f)$ durchgeführt werden.

$$Q(f) = \begin{cases} 1 & \text{für } f \in [-f_g, f_g] \\ 0 & \text{sonst} \end{cases} \quad (\text{Gl. 4.24})$$

Der Rechteckfunktion $Q(f)$ im Frequenzbereich entspricht eine $si(x)$ Funktion im Zeitbereich.

$$q(t) = 2f_g \cdot si(2\pi f_g t) \overset{\circ}{\longleftrightarrow} Q(f) \quad (\text{Gl. 4.25})$$

Die Multiplikation im Frequenzbereich kann durch eine Faltung im Zeitbereich realisiert werden. Die kontinuierliche Reflektionskurve $r(t)$ läßt sich daher auch als Faltung der Abtastwerte mit der Funktion $q(t)$ darstellen:

$$r(t) = r_a(t) * q(t) \quad (\text{Gl. 4.26})$$

Die kontinuierliche Reflektionskurve kann als Summe von $si(x)$ Funktionen betrachtet werden, die mit den jeweiligen Abtastwerten bewertet werden.

$$\begin{aligned} r(t) &= \left(\sum_{i=1}^m r(iT_s) \delta(t - iT_s) \right) * f_s si(\pi f_s t) \quad \text{mit } f_s = \frac{f_g}{2} \\ &= f_s \sum_{i=1}^m r(iT_s) si(\pi f_s (t - iT_s)) \end{aligned} \quad (\text{Gl. 4.27})$$

Das so gewonnene kontinuierliche Signal wird erneut, mit einem um den Faktor a korrigierten Abtastzeitintervall T_s' , abgetastet. Der Korrekturfaktor a wird durch Bestimmen des Maximums des folgenden Integrales bestimmt:

$$I(a) = \int_0^{mT_s} r_{\text{Soll}}(t) \cdot r_{\text{Ist}}(a \cdot t) dt \quad (\text{Gl. 4.28})$$

Weichen die Abtastfrequenzen, mit denen der Sollmusterdatensatz und der Istmusterdatensatz bestimmt wurden, zu sehr voneinander ab, dann kann ein anderes lokales Maximum gefunden werden, welches zu einem völlig falschen Ergebnis beim Vergleich der beiden Musterdatensätze führt. In der Praxis zeigte sich, daß der quarzstabilisierte Oszillator der Zeitbasis der Reflektometerbaugruppe einen so geringen Frequenzjitter aufweist, daß eine Korrektur nach dem oben angegebenen Verfahren nicht notwendig ist.

Die Verstärkungsschwankungen des Eingangverstärkers werden durch eine Amplitudenormierung ausgeglichen. Unter der Annahme, der Reflektionsverlauf habe sich nicht verändert, ist die Energie des Reflektionssignals ebenfalls konstant geblieben. Proportional der Energie des Reflektionssignales ist die Größe E :

$$E = \sum_{i=1}^m r_i^2 \quad (\text{Gl. 4.29})$$

Durch Division der einzelnen Musterwerte r_i eines Musterdatensatzes durch die Quadratwurzel von E können Verstärkungsschwankungen eliminiert werden.

Das Verfahren zur Korrektur des durch Verstärkungsschwankungen verursachten Fehlers kann auch als Normierung des Vektors \underline{r} aufgefaßt werden. E ist dann gleich dem Betragsquadrat des Vektors \underline{r} .

Der Fehler, der durch eine Offset-Spannung entsteht, kann durch Entfernen des Gleichanteils aus dem Musterdatensatz eliminiert werden. Der Gleichanteil G entspricht dem Mittelwert der Abtastwerte einer Reflektionskurve:

$$G = \frac{1}{m} \sum_{i=1}^m r_i \quad (\text{Gl. 4.30})$$

Der Mittelwert G wird von jedem Musterwert subtrahiert. Man erhält damit einen gleichanteil-freien Musterdatensatz.

Dem Reflektionssignal wird Rauschen überlagert. Der durch das Rauschen verursachte Fehler wird durch die Mittelwertbildung aus n Messungen für einen Musterwert s_i des Sollmusterdatensatzes oder i_i des Istmusterdatensatzes bereits genügend eliminiert.

Die wesentlichen Meßfehler, die bei einer Reflektionsmessung auftreten, können durch die oben beschriebenen Verfahren eliminiert werden.

Für die im Anschluß vorgestellten Mustererkennungsverfahren werden nach den oben beschriebenen Verfahren Soll- und Istmusterdatensätze \underline{s} , \underline{i} berechnet. Aus n Messungen werden durch Mittelwertbestimmung nach Gl. 4.19 die einzelnen Meßwerte eines Musterdatensatzes bestimmt.

Die n Messungen werden unmittelbar nacheinander durchgeführt. Der Fehler, der durch einen Jitter der Abtastfrequenz verursacht wird, ist bei unmittelbar aufeinanderfolgend durchgeführten Messungen sehr gering. Eine Mittelung der einzelnen Abtastwerte ist daher zulässig. Für einen Mustervergleich wird der Musterdatensatz auf den Betrag 1 normiert. Der nach Gl. 4.30 bestimmte Gleichanteil wird abschließend von jedem Musterwert subtrahiert. Der so bestimmte Istmusterdatensatz \underline{i} wird dann mit dem Sollmusterdatensatz \underline{s} verglichen, der nach der Installation des Netzsegmentes gemessen wurde. Der Sollmusterdatensatz \underline{s} wird nach dem selben Verfahren erstellt wie ein Istmusterdatensatz.

4.5.3.2 Mustererkennung durch Bestimmung des Korrelationskoeffizienten

Ein Maß für die Ähnlichkeit von \underline{r}_1 und \underline{r}_2 ist der Korrelationskoeffizient k :

$$k = \frac{\sum_i (r_{1i} - G_{r1})(r_{2i} - G_{r2})}{\sqrt{\sum_i (r_{1i} - G_{r1})^2} \sqrt{\sum_i (r_{2i} - G_{r2})^2}} \quad (\text{Gl. 4.31})$$

G_{r1} und G_{r2} sind die Mittelwerte der Musterdatensätze r_1 , r_2 . Der Korrelationskoeffizient k ist identisch mit der diskreten Kreuzkorrelation an der Stelle $j=0$.

$$r_{r_2}(j) = \sum_i r'_{1i} \cdot r'_{2(i+j)} \quad (\text{Gl. 4.32})$$

Unter der Voraussetzung, daß die Musterdatensätze r_1 und r_2 so normiert werden, daß sie den Mittelwert Null und den Betrag 1 haben (Gl. 4.33).

$$r'_i = \frac{r_i - G}{\sqrt{\sum_i (r_i - G)^2}} \quad (\text{Gl. 4.33})$$

Da die Musterdatensätze s und i bei ihrer Erstellung bereits auf den Betrag 1 normiert und vom Gleichanteil befreit werden, ergibt sich aus Gl. 4.31 für die Bestimmung des Korrelationskoeffizienten:

$$k = \sum_{i=1}^m s_i \cdot i_i \quad (\text{Gl. 4.34})$$

Der Korrelationskoeffizient ist gleich 1 genau dann, wenn die beiden Signale identisch sind (und nicht verschwinden). Es läßt sich damit eine Regel für einen Vergleich angeben:

$$\begin{aligned} k \in (k_{th}, 1] & \text{ Übereinstimmung} \\ k \in [-1, k_{th}] & \text{ keine Übereinstimmung} \end{aligned} \quad (\text{Gl. 4.35})$$

wobei k_{th} ein geeignet gewählter Schwellwert nahe 1 ist. Wegen der unter 4.5.3.1 besprochenen Meßfehler ist der Korrelationskoeffizient k nahezu immer kleiner als 1. Erfahrungswerte für k_{th} liegen bei $k_{th} = 0,985$.

Der Korrelationskoeffizient k kann auch als das skalare Produkt zweier normierter Vektoren r_1 und r_2 interpretiert werden.

$$k = s \cdot i \quad (\text{Gl. 4.36})$$

Betrachtet man die beiden zu vergleichenden Musterdatensätze als normierte Vektoren im n -dimensionalen Raum, so sind diese genau identisch, wenn ihr skalares Produkt den Wert 1 annimmt. Stehen die Vektoren orthogonal aufeinander, so nimmt das skalare Produkt den Wert Null an. Haben die beiden Vektoren genau die entgegengesetzte Richtung, dann hat das skalare Produkt den Wert -1 .

Um den Ort des Fehlers bestimmen zu können, unterteilt man die beiden Datensätze in Teilmusterdatensätze der Werteanzahl l . Bild 4.19 erläutert die Unterteilung eines Musterdatensatzes.

Die gebildeten Teilmusterdatensätze überlappen sich um die Werteanzahl u . Übliche Werte für Teilmusterdatensätze sind $l = 5$ und $u = 2$. Teilmusterdatensätze werden auch als Teilvektoren bezeichnet. Die Mustererkennung wird nun für jedes einander zugeordnete Paar der Teilmusterdatensätze aus \underline{g} und \underline{i} durchgeführt. Eine Veränderung der elektrischen Eigenschaften des Netzsegmentes liegt dann vor, wenn die Teilvektoren nicht mehr übereinstimmen. Ist i die Ordnungsnummer der Teilmusterdatensätze, die nicht mehr übereinstimmen, so ergibt sich für den Abstand l_f des Fehlerortes vom Meßort unter Einbeziehung von Gl. 4.17:

$$l_f = i \cdot (l - u) \cdot \Delta s \tag{Gl. 4.37}$$

$$\approx i \cdot (l - u) \cdot 57,75 \text{cm}$$

bei einem maximalen Fehler von $\pm (l - u) \cdot \Delta s$

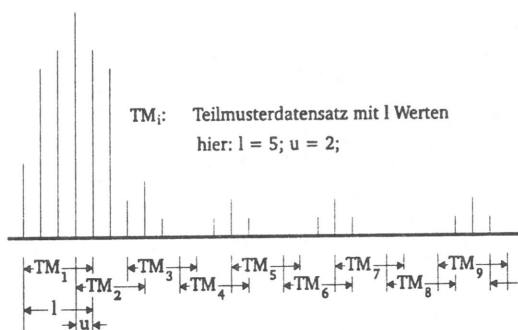


Bild 4.19 Unterteilung eines Musterdatensatzes in Teilmusterdatensätze

4.5.3.3 Mustererkennung mit neuronalen Netzen

Betrachtet ein menschlicher Experte eine Reflektionskurve, so erkennt er sofort, ob deren Verlauf akzeptiert werden kann oder nicht. Der Mensch ist in der Lage, in Bruchteilen einer

Sekunde komplizierte Signale (Bilder, Lautfolgen usw.) zu erkennen und inhaltlich zu interpretieren. So erkennt der Mensch z.B. Menschen wieder, die er lange nicht gesehen hat und die sich in der Zwischenzeit sehr verändert haben. Andere Bereiche dagegen fallen dem Menschen schwer, z.B. wenn er schnell und exakt Quadratwurzeln aus beliebigen Zahlenwerten ziehen soll. Letzteres fällt einem Rechner konventioneller Bauart sehr leicht, während Mustererkennung mit diesem nur sehr schwer durchzuführen ist.

Die Grund dafür, daß das menschliche Gehirn wesentlich schneller Muster erkennen kann, liegt nicht in der Verarbeitungsgeschwindigkeit oder Genauigkeit der durchgeführten Operationen, die mit den vorliegenden Informationen durchgeführt werden, sondern in der hochgradig parallelen Natur der Informationsverarbeitung im Gehirn.

Das Gehirn enthält bei einem durchschnittlichen Gewicht von 1500 Gramm zwischen 10 und 100 Milliarden Neuronen, die in einer netzwerkartigen Struktur miteinander verbunden sind.

Die Struktur des menschlichen Gehirns läßt sich in kleinen Ausschnitten, in sogenannten neuronalen Netzen, modellieren und auf konventionellen Rechnern simulieren [131, 132, 161, 175]. Für das hier vorliegende Problem: "Mustererkennung eines Datensatzes mit maximal 866 Einzelwerten" wurde die Anwendung simulierter Netzmodelle für die Mustererkennung untersucht.

Die Verarbeitungselemente eines neuronalen Netzes sind die Neuronen. Bild 4.20 zeigt das Modell eines Neurons.

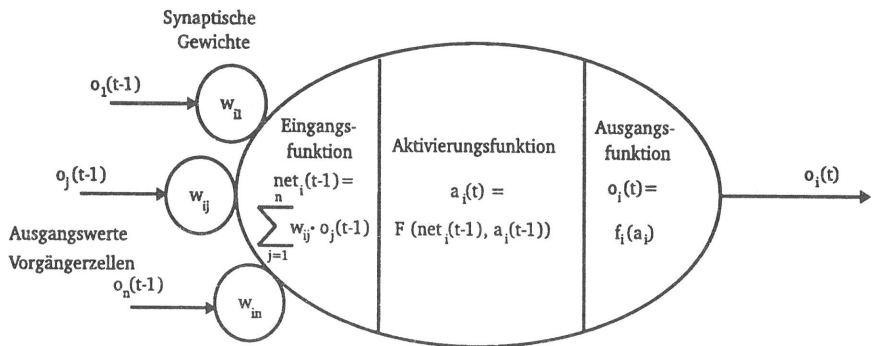


Bild 4.20 Modell eines Neurons

Das Neuron-Modell kann in drei Bereiche unterteilt werden:

- Eingangsfunktion

- Aktivierungsfunktion
- Ausgangsfunktion

Die Eingangswerte eines Neurons werden mit den synaptischen Gewichten w_{ij} bewertet. Der Index ij deutet darauf hin, daß der Ausgangswert des Neurons i mit einem Eingang des Neurons j verbunden ist. Die Eingangsfunktion des Neurons bestimmt aus den gewichteten Eingangswerten einen Wert, der, neben dem alten Zustand des Neurons, als Argument für die Aktivierungsfunktion dient. In Bild 4.20 wird als Eingangsfunktion eine Aufsummierung der gewichtete Eingangswerte verwendet. Beispiele für die Aktivierungsfunktion sind:

- Sprungfunktion,
- Signumfunktion,
- Sigmoidfunktion
- u.a.

Der Wert der Aktivierungsfunktion wiederum ist das Argument der Ausgangsfunktion. Der durch die Ausgangsfunktion ermittelte Wert wird an weitere Neuronen übergeben und dort als Eingangswert weiterverarbeitet.

Mit den Neuronen lassen sich unterschiedlichste Netzstrukturen realisieren. Man unterscheidet zwischen rückgekoppelten (feed back) und nicht rückgekoppelten (feed forward) neuronalen Netzen. Eine Einteilung der unterschiedlichen Netztypen findet man in [161, 175]

Die Information, die notwendig ist, um ein Muster zu erkennen, ist in den Gewichten w_{ij} enthalten. Die Bestimmung der Gewichte w_{ij} erfolgt während der Lernphase durch wiederholte Anpassung nach Lernregeln. Während der Anwendung des neuronalen Netzes findet im allgemeinen kein Lernen statt.

Für die Anwendung zur Mustererkennung bei Musterdatensätzen, die durch Reflektionsmessungen gewonnen wurden, wurde eine mehrlagige (multi layer) Feed Forward Netzstruktur gewählt [35]. Bild 4.21 zeigt die Struktur eines solchen neuronalen Netzes. Diese Netzarchitektur wird in der Fachliteratur auch als mehrlagiges Perceptron bezeichnet. Als Lernregel wird die Back-Propagation-Lernregel verwendet [161, 175]. Während des Lernens werden dem Netz am Eingang Musterdatensätze präsentiert. Am Ausgang des Netzes wird der durch das neuronale Netz ermittelte Ausgangswert mit dem gewünschten Ausgangswert (Zielwert) verglichen. Ein Lernzyklus besteht aus 2 Phasen:

- Forward-Propagation: Mit den momentanen Gewichten w_{ij} wird ein Ausgangswert berechnet.
- Die Differenz zwischen Ausgangswert und Zielwert wird zur Berechnung von Korrekturen für die Gewichte w_{ij} verwendet. Die Berechnung der Korrekturen beginnt für die Gewichte

w_{ij} zwischen der letzten Lage und dem Ausgang und endet für die Gewichte zwischen Eingang und der ersten Lage des neuronalen Netzes.

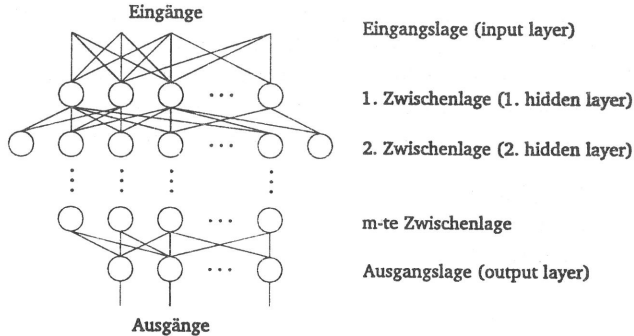


Bild 4.21 Neuronales Netz mit Feed Forward Netzstruktur. Bei diesem Netztyp wird keiner der Ausgänge eines Neurons in eine vorhergehende Netzlage zurückgekoppelt.

Für die Mustererkennung wird der Musterdatensatz, wie in Bild 4.19 gezeigt, in Teilmusterdatensätze der Werteanzahl l unterteilt. Für jeden Teilmusterdatensatz werden durch einen Lernvorgang die Gewichte w_{ij} für ein entsprechendes neuronales Netz bestimmt. Dadurch kann die Netzarchitektur auf ein Netz mit l Eingängen und einem Ausgang beschränkt werden. Der Fehlerort wird dann durch die Ordnungsnummer des neuronalen Netzes bestimmt, das keine Übereinstimmung des angebotenen Teilmusterdatensatzes mit dem gelernten Teilmusterdatensatz des Sollmusterdatensatzes anzeigt.

Die Anzahl der notwendigen Lagen (layer) des neuronalen Netzes läßt sich durch die Komplexität der zu erkennenden Muster bestimmen. Im Fall der mehrlagigen Feed Forward Netzstruktur können 3 Fälle unterschieden werden:

- **keine Zwischenlage:**

Das Netz kann den l -dimensionalen Raum, der durch die Anzahl der Werte des Teilmusterdatensatzes definiert wird, in 2 Teilräume unterteilen. Die Abgrenzung der Teilräume erfolgt durch eine Ebene der Dimension $l-1$.

- **eine Zwischenlage:**

Das Netz kann den l -dimensionalen Eingangsraum in zwei Teilräume unterteilen, deren Begrenzungsfläche offen oder geschlossen sein kann, aber konvex sein muß (nur eine Krümmungsrichtung ist zugelassen).

zwei Zwischenlagen:

Das Netz kann im 1-dimensionalen Eingangsraum beliebige Gebiete abgrenzen. Die einzige Einschränkung ist die Anzahl der Neuronen in den Zwischenlagen. Bei genügend hoher Anzahl können auch komplexere Gebiete eingeschlossen werden.

Bild 4.22 zeigt am Beispiel eines 2-dimensionalen Eingangsraumes die Möglichkeiten der Abgrenzung von Gebieten.

Mit steigender Anzahl von Zwischenlagen steigt auch die Dauer der Lernphase an. Für die Mustererkennung bei Reflektionsmessungen erwies sich ein neuronales Netz mit einer Zwischenlage als ausreichend. Für die Lernphase wurden als Gutmuster aktuell gemessene Musterdatensätze verwendet. Als Schlechtmuster wurden spezielle Musterdatensätze durch Einbau von Fehlern am betreffenden Netzsegment, oder durch Addition von Rauschen und Störspitzen, erzeugt.


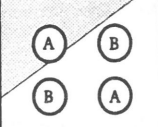

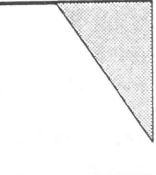
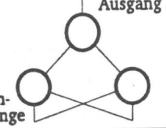
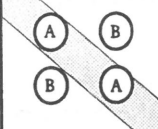

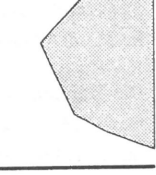
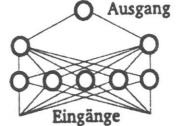
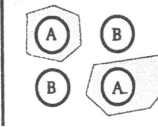
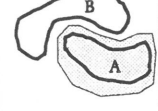
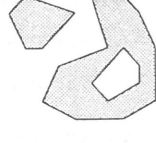
Neuronale Netzstruktur	Typ des Entscheidungsgebiets	exklusiv-ODER Problem	Vermaschte Gebiete	Allgemeinste Gebietsform
<p>eine Ebene</p>  <p>Ausgang</p> <p>Eingänge</p>	<p>Halbebene begrenzt durch Hyperebene</p> <p>(In der Abbildung eine Gerade)</p>			
<p>zwei Ebenen (1 hidden layer)</p>  <p>Ausgang</p> <p>Eingänge</p>	<p>Konvexe offene oder geschlossene Gebiete</p>			
<p>drei Ebenen (2 hidden layer)</p>  <p>Ausgang</p> <p>Eingänge</p>	<p>Beliebig</p> <p>Komplexität durch Anzahl der Zellen beschränkt</p>			

Bild 4.22 Auswirkung unterschiedlicher Anzahl von Zwischenlagen (hidden layer) bei neuronalen Netzen mit Feed Forward Netzstruktur.

Neuronale Netze sind für die Mustererkennung bei solch kleinen Datensätzen, wie sie bei vorgestellten Reflektionsmessung entstehen, durchaus geeignet. Ein Problem stellt jedoch die

relativ lange Lernphase dar [35, 127]. Nach erfolgtem Lernen erfolgt die Mustererkennung jedoch sehr schnell.

An Algorithmen zur Beschleunigung der Back-Propagationen-Lernregel wird gearbeitet. An dieser Stelle sei auf die entsprechende Literatur verwiesen [43, 163].

4.5.4 Weitere Systemerweiterungen

Für das Meßsystem wurden, neben den detailliert vorgestellten Meßbaugruppen, weitere Erweiterungsbaugruppen realisiert. Hierzu zählen:

- Floppy-Disk Baugruppe
- Speichererweiterungsbaugruppe
- Uhrenbaugruppe

Die Fernsteuerung der Meßstationen erfolgt über dasselbe lokale Rechnernetz, das mit dem Meßsystem auch beobachtet werden soll (inbound). Um dem Verlust von Meßdaten bei einem totalen Netzausfall vorzubeugen, wurde ein nichtflüchtiger Massendatenspeicher in Form einer Floppy-Disk Baugruppe [139] für die Meßstationen realisiert. Auf der Floppy-Disk Baugruppe befindet sich ein 3,5" Floppy-Disk Laufwerk. Die Meßdaten werden auf der Floppy-Disk im UNIX Standard tar-Format gespeichert. Der Datenträger kann auf jedem PC ausgelesen und interpretiert werden. Die Speicherung eines Meßergebnisses auf der Floppy-Disk muß im Meßauftrag angegeben werden.

Wie unter 4.3.1 beschrieben, verfügt die Basisbaugruppe nur über einen Speicherausbau von 128 KByte RAM und 128 KByte ROM. Der verwendete Prozessor i80186 kann 1 MByte adressieren. Die Speichererweiterungsbaugruppe stellt für den restlichen Adressraum wahlweise RAM oder ROM zur Verfügung.

Die Uhrenbaugruppe erweitert das Gesamtsystem um eine batteriegepufferte Echtzeituhr und max 128 KByte batteriegepuffertes RAM. In dem batteriegepufferten RAM-Bereich werden nichtflüchtig Adresstabellen für Echotests und sonstige Konfigurationsdaten gespeichert. Ferner werden zusätzliche parallele und serielle Schnittstellen zur Verfügung gestellt.

4.6 Systemsoftware

4.6.1 Betriebssystem

Als Betriebssystem für die Meßstationen wurde ein meldungsgesteuertes Pseudo-Multitasking-Betriebssystem entwickelt, welches aufgrund seiner internen Meldungswarteschlange als Queue Operating System (QOS) bezeichnet wird [49, 138, 140, 168]. Bild 4.23 erläutert das Prinzip von QOS.

Das Zusammenwirken aller Module (Systemprozesse und Anwendungen) erfolgt über den Austausch von Meldungen. Die zentrale Meldungswarteschlange nimmt alle Meldungen auf. Die

dort gespeicherten Meldungen werden von einem zentralen Verteilermodul den im System registrierten Modulen angeboten.

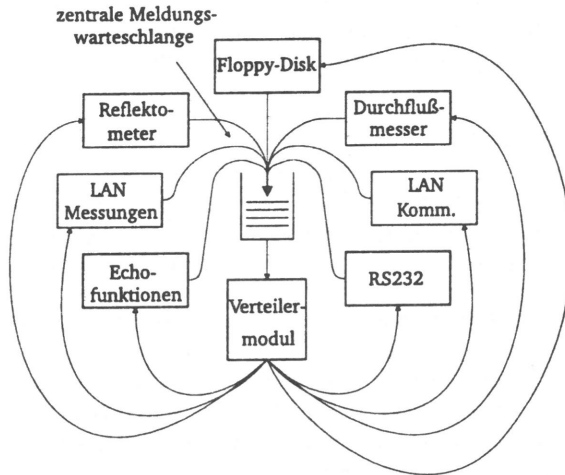


Bild 4.23 Prinzip des Queue Operating Systems der Meßstationen

Die Meldungen werden von den Modulen vollständig bearbeitet, bevor sie die Ablaufkontrolle an das Verteilermodul zurückgeben. Alle Eingaben an ein Modul werden in die Meldungs-warteschlange eingetragen. Die Prozeßumschaltung erfolgt also nicht nach dem Time-sharing Verfahren, bei dem jedem Anwendungsprozeß ein bestimmter Zeitschlitz zur Verfügung gestellt wird, sondern meldungsgesteuert. Zur Vermeidung unnötiger Prozeßwechsel melden die einzelnen Module sich beim Verteilermodul für einen bestimmten Meldungstyp an.

Eine Ausnahme von der normalen Prozeßumschaltung bilden asynchrone Ereignisse, die von Schnittstellen und Timern kommen. Sie werden nicht vom Verteilermodul abgefragt, sondern lösen Unterbrechungen aus. Im Falle einer Unterbrechung besteht die Möglichkeit, in der Bearbeitungsroutine der Unterbrechung Meldungen zu generieren oder die Bearbeitung direkt durchzuführen.

Das gewählte Betriebssystem hat den Nachteil, daß die einzelnen Module nicht durch eine Instanz des Betriebssystems unterbrochen werden können. Es muß daher bei der Implementierung von Anwendungen und Systemprozessen streng darauf geachtet werden, daß diese nach der notwendigen Bearbeitungszeit die Kontrolle wieder an das Verteilermodul zurückgeben. Sorgfältige Tests der einzelnen Module waren daher unumgänglich.

Wesentlicher Vorteil des Betriebssystems ist dessen Einfachheit und dessen Modularität.

4.6.2 Kommunikationssystem

4.6.2.1 Anforderungen an das Kommunikationssystem

Die Meßstationen und die Monitoring- und Diagnosestation kommunizieren über das lokale Rechnernetz mittels eines einfachen Kommunikationsprotokolls [39, 50, 69, 140]. Die Schichtung des Kommunikationsprotokolls zeigt Bild 4.24.

Die Schichten 1 und 2a des Kommunikationsprotokolls entsprechen dem Standard für lokale Netze mit CSMA/CD Zugriffsverfahren IEEE 802.3. Die Schicht 2b wird konform dem Standard IEEE 802.2 Logical Link Control (LLC) realisiert. Die von Schicht 2b angebotenen Dienste werden von einem spezifischen Protokoll verwendet, welches als Low Level Transport Protocol (LLTP) bezeichnet wurde. Auf das LLTP setzen die einzelnen Anwendungen auf. Diese Schicht wurde als Measurement Application Service Protocol (MASP) bezeichnet.

Große lokale Rechnernetze werden durch Netzkoppeleinheiten in Teilnetze aufgeteilt. Die Netzkoppeleinheiten verbinden die Teilnetze über die Vermittlungsschicht (Schicht 3) der verwendeten Protokolle. Solche Netzkoppeleinheiten werden auch als Router bezeichnet. Die Meßsystem-Protokoll-Architektur verfügt über keine Vermittlungsschicht. In jedem Teilnetz muß daher eine Monitoringstation vorgesehen werden. Die Monitoringstationen untereinander kommunizieren dann über ein Standard Protokoll (z.B. TCP/IP). Diese Hierarchisierung hat zwei Vorteile. Zum einen wird der vom verteilten Meßsystem generierte Datenverkehr auf ein Teilnetz beschränkt, zum anderen wird die Meßsystem-Protokoll-Architektur, und damit auch deren Implementierung, einfacher.

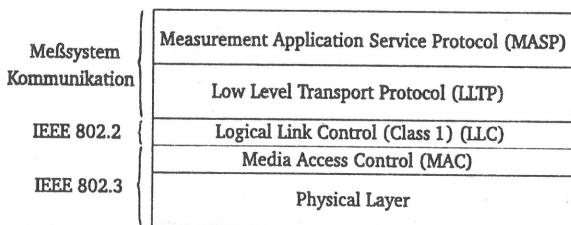


Bild 4.24 Schichtung der Meßsystem Protokoll-Architektur

Die wesentlichen Funktionalitäten des LLT-Protokolls sind:

- Bereitstellen einer gesicherten Datenübertragung.
- Es können mehrere Verbindungen verwaltet werden. Die Unterscheidung der einzelnen Verbindungen erfolgt durch Anwendungszugangspunkte, den sogenannten Application Access Points (AAP).
- Für jeden AAP auf einer Meßstation kann es nur eine Verbindung geben.

- Die LLTP-Instanz der Monitoring- und Diagnosestation kann anhand eines AAP mehrere Verbindungen verwalten. Die Verbindungen werden anhand der Media Access Control (MAC) Adressen unterschieden.
- Für die Meßapplikationen auf den Meßstationen werden feste AAPs vergeben.
- Für Rückmeldungen der Meßstationen an das Monitoringsystem sind ebenfalls bestimmte AAPs reserviert.
- Die Verwendung von Gruppenadressen ist möglich.
- Der Verbindungsaufbau ist sowohl von der Monitoringstation als auch von der Meßstation aus möglich.
- Der Verbindungsabbau erfolgt durch den Kommunikationspartner, der die Verbindung aufgebaut hat. Die LLTP-Instanz der zentralen Monitoring- und Diagnosestation darf auch nicht von ihr aufgebaute Verbindungen schließen.

4.6.2.2 Das Low Level Transport Protokoll (LLTP)

Innerhalb der LLTP Instanz werden die einzelnen Verbindungen durch:

- Source Application Access Point (SAAP)
- Destination Application Access Point (DAAP)
- Media Access Control Source Address (SA)
- Media Access Control Destination Address (DA)

unterschieden. Den Anwendungen auf den Meßstationen werden netzweit dieselben AAPs zugeordnet. So hat z.B. die Anwendung Reflektionsmessung immer den AAP 10.

Wird vom Monitoringsystem eine Messung initiiert, dann wird zur Durchführung der Messung ein eigener Meßprozeß gestartet. Dieser Meßprozeß meldet sich bei der LLTP-Instanz auf der Monitoring- und Diagnosestation unter Angabe eines AAP an. Die AAPs auf der Monitoring- und Diagnosestation werden dynamisch vergeben. Bestimmte Verbindungen sind reserviert für den Aufbau von Verbindungen ausgehend von den Meßstationen (z.B. für Remote Login, Alarmmeldungen usw.). Ein Meßprozeß kann mehrere Verbindungen öffnen. Bild 4.25 zeigt ein Kommunikationsszenario.

Ein Meßprozeß wird vom Monitoringsystem auf der Monitoring- und Diagnosestation gestartet. Um mit einer Meßstation kommunizieren zu können, meldet sich der Meßprozeß mit einem Open Request unter Angabe eines beliebigen AAPs bei der LLTP-Instanz an. Ist der angegebene AAP frei, wird die Anmeldung mit der Meldung Open Confirm quittiert. Ist der angegebene AAP nicht frei, so wird eine Fehlermeldung zurückgegeben.

Nach erfolgreicher Anmeldung können Daten übertragen werden. Mit der ersten Datenübertragung von der Monitoring- und Diagnosestation zur Meßstation wird gleichzeitig die Verbindung aufgebaut. Jedes übertragene Datenpaket wird nummeriert und entsprechend quittiert.

Die Meßstation überträgt die Meßergebnisse über diese Verbindung an die LLTP-Instanz der Monitoring- und Diagnosestation.

Nach erfolgreicher Datenübertragung wird, ausgehend von der LLTP-Instanz der Monitoring- und Diagnosestation, die Verbindung wieder abgebaut. Dazu sendet die LLTP-Instanz eine Disconnect Request-Meldung.

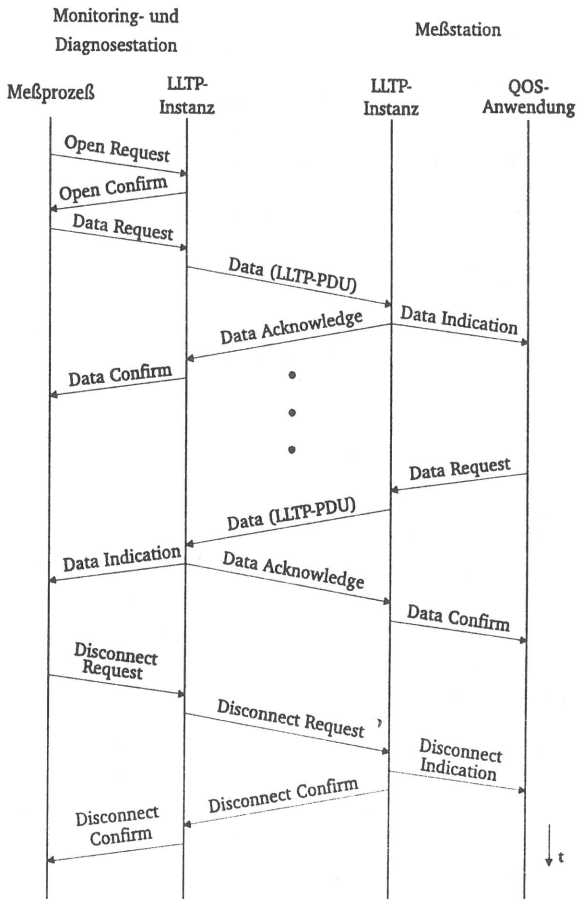


Bild 4.25

Kommunikationsszenario für die Kommunikation zwischen den LLTP-Instanzen der Monitoring- und Diagnosestation und einer Meßstation.

Eine Anwendung auf der Meßstation, die versucht auf einer abgebauten Verbindung zu senden, erhält einer Fehlermeldung. Sie kann dann über einen Fehlerkanal Verbindung mit der LLTP-Instanz der Monitoring- und Diagnosestation aufnehmen.

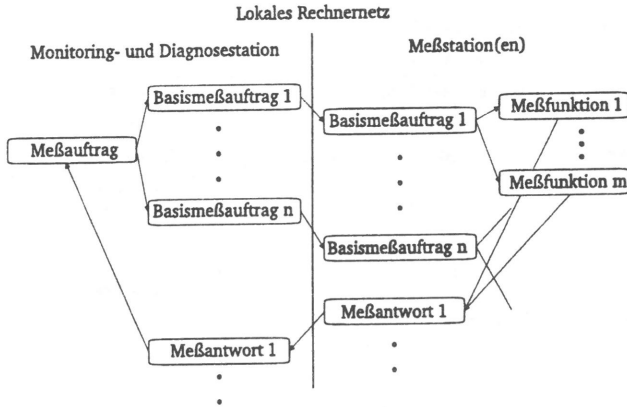


Bild 4.26 Erläuterung der Begriffe Meßauftrag, Basismeßauftrag, Meßfunktion und Meßantwort.

Ein Meßprozeß unterteilt einen Meßauftrag in sogenannte Basismeßaufträge (Bild 4.26). Basismeßaufträge sind die elementaren Meßaufträge, die von einer Meßstation durchgeführt werden können. Die Basismeßaufträge werden über das LLTP zu den Meßstationen übertragen. Ein Basismeßauftrag wird in einer Meßstation in Meßfunktionen unterteilt. Die Meßfunktionen können direkt von der Basisbaugruppe oder einer Erweiterungsbaugruppe durchgeführt werden. Ein Meßauftrag besteht aus mindestens einem Basismeßauftrag, ein Basismeßauftrag aus mindestens einer Meßfunktion. Das Ergebnis eines Basismeßauftrages wird als Meßantwort an das Monitoringsystem übertragen.

Neben den Basismeßaufträgen sind Konfigurationsaufträge für die Initialisierung der Meßstationen vorgesehen.

Die über das lokale Rechnernetz ausgetauschten Protokolldateinheiten (PDUs, Protocol Data Units) haben das in Bild 4.27 dargestellte Format.

Der Aufbau der Protokolldateinheiten der Schichten unterhalb der LLTP Schicht entspricht den IEEE Standards IEEE 802.3 und IEEE 802.2. Am Anfang der LLTP-Protokolldateinheit wird in 2 Bytes der Ziel-Anwendungszugangspunkt (Destination Application Access Point, DAAP) angegeben. Dem DAAP folgt, ebenfalls 2 Bytes lang, der Quell-Anwendungs-

zugangspunkt (Source Application Access Point, SAAP). Bild 4.27 zeigt den Aufbau der LLTP-PDU und deren Einbau in die MAC-PDU.

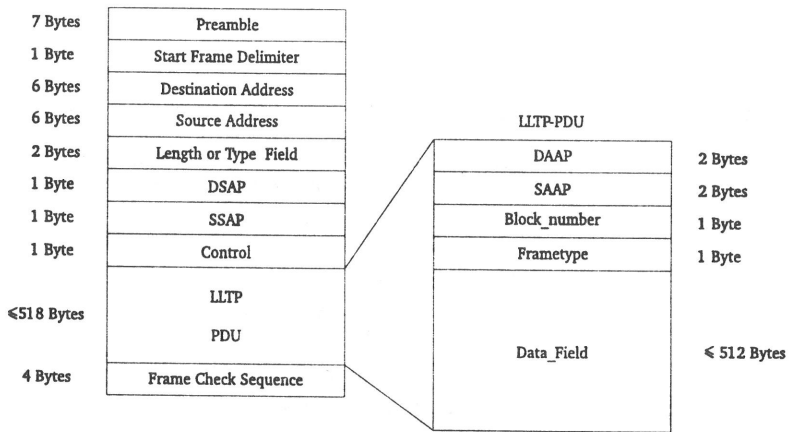


Bild 4.27 Aufbau einer Protokolldateieneinheit der MAC-Schicht mit eingebauter LLTP-Protokolldateieneinheit

Aufeinanderfolgende Pakete einer Verbindung werden Modulo 256 durchnummeriert. Die Paketnummer wird im Block_number Feld übertragen.

Es werden folgende Typen von LLTP-PDUs unterschieden:

- Data Request
- Data Acknowledge
- Disconnect Request
- Disconnect Acknowledge

Für den Verbindungsaufbau sind keine speziellen LLTP-PDU-Typen notwendig, da der Verbindungsaufbau mit der ersten Datenübertragung erfolgt.

4.6.3 Anwendungen

Funktionen funktioneller Einheiten der Meßstation werden unter dem Begriff Anwendung zusammengefaßt. Folgende Anwendungen werden unterschieden:

- Meßfunktionen der Reflektometerbaugruppe (AAP: 10)
- Meßfunktionen der Durchflußmesserbaugruppe (AAP: 11)
- Meßfunktionen der Basisbaugruppe (AAP: 12)
- Datensicherung auf der Floppy-Disk (AAP: 13)

- Systemanwendungen (Verteilmodul, LLTP-Instanz, RS232-Treiber...)

Die Anwendungen realisieren die Meßmöglichkeiten der Meßstationen.

5. Das Monitoringsystem

Die im lokalen Rechnernetz verteilten Meßstationen werden zentral vom Monitoringsystem gesteuert. Die Steuerung erfolgt über das unter 4.2.2 beschriebene Kommunikationssystem. Bild 5.1 zeigt die Architektur des Monitoringsystems [24, 126, 170].

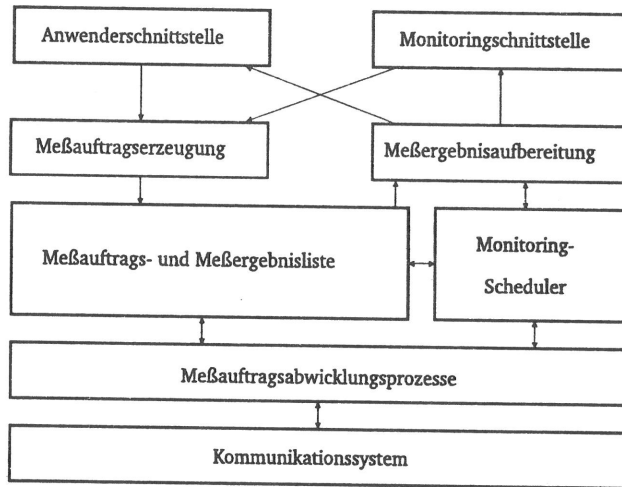


Bild 5.1 Architektur des Monitoringsystems

Die Meßauftragserzeugung des Monitoringsystems kann sowohl über die Anwenderschnittstelle als auch über die Monitoringschnittstelle angestoßen werden. Von der Meßauftragserzeugung werden Meßaufträge in die Meßauftrags- und Meßergebnisliste eingetragen. Zu einem Meßauftrag gehören Parameter wie:

- Startzeitpunkt der Messung
- Dauer der Messung / Endzeitpunkt der Messung
- Art der Ergebnisaufbereitung
- usw.

Kern des Monitoringsystems ist der sogenannte Monitoring-Scheduler. Der Monitoring-Scheduler überwacht laufend die Einträge in der Meßauftrags- und Meßergebnisliste und startet zum gewünschten Zeitpunkt einen Meßprozeß, der den Meßauftrag durchführt. Der Meßprozeß führt die Unterteilung des Meßauftrags in Basismeßaufträge durch.

Die Meßprozesse kommunizieren über die LLTP-Instanz der Monitoring- und Diagnosestation mit dem verteilten Meßsystem. Die empfangenen Meßergebnisse werden gespeichert, und das Vorliegen der Meßergebnisse wird in der Meßauftrags- und Meßergebnisliste vermerkt. Der Monitoring-Scheduler stößt nach Vorliegen aller Ergebnisse eines Meßauftrags die Meßergebnisaufbereitung an. Die Meßergebnisaufbereitung orientiert sich an den Einträgen in der Meßauftrags- und Meßergebnisliste. Die gespeicherten Meßergebnisse werden entsprechend verarbeitet, und das Resultat der Bearbeitung wird auf die gewünschte Schnittstelle ausgegeben. Der abgearbeitete Meßauftrag wird nach vollständiger Bearbeitung aus der Meßauftrags- und Meßergebnisliste gelöscht.

Die Architektur des Monitoringsystems wurde so gewählt, daß neben dem verteilten Meßsystem auch andere Meßkomponenten integrierbar (z.B. Protokollanalytoren) sind. Voraussetzung ist deren Fernsteuerbarkeit.

Die weitere Entwicklung des Monitoringsystems war nicht Gegenstand dieser Arbeit und wird gegenwärtig am IND weiterverfolgt [172].

6. Ein Diagnosesystem für die wissensbasierte Diagnose in komplexen Systemen

6.1 Architektur des Diagnosesystems

Das Diagnosesystem soll den Anwender von lokalen Rechnernetzen bei der Fehlersuche unterstützen. Bild 6.1 zeigt die Architektur des Diagnosesystems.

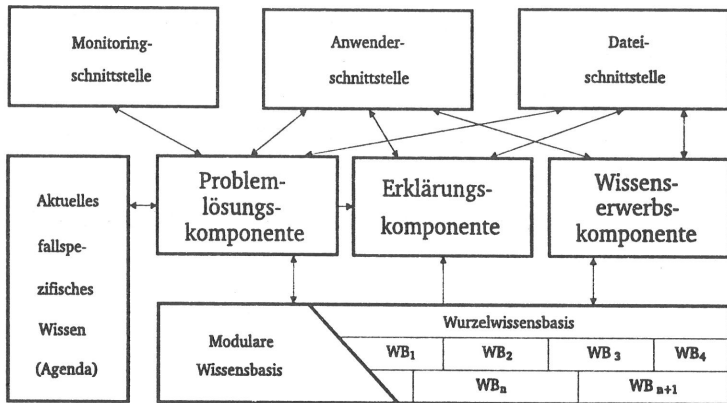


Bild 6.1 Architektur des Diagnosesystems

Das Diagnosesystem [2, 19, 176, 121, 155] verfügt über drei Schnittstellen für den Informationsaustausch:

- Anwenderschnittstelle
- Monitoringschnittstelle
- Dateischnittstelle

Auf die Anwendung der einzelnen Schnittstellen wird in den Beschreibungen der Hauptkomponenten des Diagnosesystems eingegangen. Die drei Hauptkomponenten des Diagnosesystems sind:

- Problemlösungskomponente
- Erklärungskomponente
- Wissenserwerbskomponente

Das Wissen über den Aufbau des lokalen Rechnernetzes, die Eigenschaften der verwendeten Netzkomponenten und deren Fehler wird als statisches Wissen bezeichnet und modularisiert in einer Datenbank gespeichert. Diese Datenbank wird auch als modulare Wissensbasis bezeichnet.

Auf die Modularisierung der Wissensbasis wird in Abschnitt 6.3 näher eingegangen. Die Wissensdarstellung anhand von Objekten ist Gegenstand von Abschnitt 6.2.

Das vom aktuellen Fehlerfall abhängige Wissen wird als fallspezifisches Wissen bezeichnet. Das fallspezifische Wissen wird während einer Diagnosesitzung erfaßt und in der Agenda gespeichert.

Eine Diagnosesitzung wird von der Problemlösungskomponente gesteuert. Dazu greift die Problemlösungskomponente auf das Wissen zurück, das in der modularen Wissensbasis gespeichert ist. Die Agenda enthält das aktuell bearbeitete Wissen. Fragen über bestimmte Meßwerte, das Vorliegen von Symptomen usw. werden entweder über die Anwenderschnittstelle an den Benutzer, oder über die Monitoringschnittstelle an das Monitoringsystem weitergegeben. Der aktuelle Stand einer Diagnosesitzung kann über die Dateischnittstelle abgespeichert und zu einem späteren Zeitpunkt wieder hergestellt werden. In Kapitel 6.5 wird auf die Architektur der Problemlösungskomponente näher eingegangen.

Die Wissenserwerbskomponente unterstützt den Anwender bei der Erstellung von Wissensbasismodulen [176, 185, 190]. Das Wissen wird anhand von Objekten in einer Wissensbeschreibungsdatei beschrieben. Über die Dateischnittstelle wird die Wissensbeschreibungsdatei von der Wissenserwerbskomponente eingelesen und interpretiert. Aus den so gewonnenen Informationen wird dann ein Wissensbasismodul generiert. Über die Anwenderschnittstelle hat der Benutzer die Möglichkeit, den Wissenserwerb zu steuern und einzelne Objekte und Objektrelationen interaktiv zu ändern. Auf den Aufbau der Wissensbeschreibungsdatei und der Wissenserwerbskomponente wird in Abschnitt 6.4 näher eingegangen.

Aufgabe der Erklärungskomponente ist es, die Vorgehensweise des Diagnosesystems dem Anwender zu erläutern. Dazu empfängt die Erklärungskomponente von der Problemlösungskomponente Meldungen, sogenannte Trace-Messages. Durch Zugriffe auf die modulare Wissensbasis können Erklärungen über den aktuellen Stand der Diagnosesitzung generiert werden. Die Erklärungen können wahlweise über die Anwenderschnittstelle auf den Bildschirm, oder über die Dateischnittstelle in ein Erklärungsfile ausgegeben werden. Auf die Architektur der Erklärungskomponente wird in Abschnitt 6.6 näher eingegangen.

6.2 Wissensdarstellung

6.2.1 Objekte der Wissensdarstellung

Das Wissen über einen Sachverhalt wird anhand folgender Objekte und Relationen zwischen diesen Objekten beschrieben [176, 185]:

- Komponenten
- Diagnosen
- Nachfolger-Wissensbasen
- Symptome
- Methoden

- Regeln
- Prämissen
- Erklärungen

Als *Komponenten* werden alle physikalischen und logischen Einheiten eines Systems bezeichnet. Anhand der Zuordnung der Komponenten untereinander wird eine Systemmodellierung für die Diagnose realisiert. Die Zuordnung von Komponenten zueinander kann prinzipiell nach zwei unterschiedlichen Ordnungsprinzipien erfolgen. Auf die beiden möglichen Ordnungsprinzipien wird in Abschnitt 6.2.2 näher eingegangen.

Die möglichen Beurteilungen einer Komponente werden als *Diagnosen* bezeichnet. Es wird zwischen *Diagnosen* und *Enddiagnosen* unterschieden. Als *Enddiagnosen* werden Beurteilungen bezeichnet, bei deren Bestätigung eine Diagnosesitzung abgeschlossen werden kann. *Diagnosen* bezeichnen Beurteilungen, bei deren Bestätigung eine Fortsetzung der Diagnosesitzung sinnvoll ist. Bei Bestätigung einer Diagnose wird entweder die zugeordnete Komponente verfeinert oder eine evtl. zugeordnete Nachfolger-Wissensbasis geladen.

Einer Komponente kann ebenfalls eine *Nachfolger-Wissensbasis* zugeordnet werden. Die zugeordnete Nachfolger-Wissensbasis dient der detaillierten Fehlersuche innerhalb der verdächtigsten Komponente. Jedes Wissensbasismodul kann als Nachfolger-Wissensbasis verwendet werden.

Erscheinungsbilder von Fehlerursachen werden als *Symptome* bezeichnet. Das fehlerabhängige Wissen wird durch die Bewertung von Symptomen erfaßt. Für den automatischen Erwerb von fallspezifischem Wissen für die Symptombewertung können *Methoden* definiert werden.

Die Beziehung von Symptomen untereinander kann in *Regeln* ausgedrückt werden. Für jedes Symptom kann eine Regel angegeben werden. Regeln orientieren sich an der Struktur der disjunktiven Normalform der booleschen Algebra. Beispiel:

Symptom 1 trifft dann zu wenn: **Symptom 2** und **Symptom 3**
 oder nicht **Symptom 4** und **Symptom 5**
 oder **Symptom 6**
 zutrifft.

Die Zuordnung von Symptomen zu *Diagnosen* und *Enddiagnosen* geschieht in *Prämissen*. *Prämissen* stellen wie *Regeln* eine Form der Verknüpfung von Symptomen dar. Jede *Prämisse* realisiert eine UND-Verknüpfung, die eine Diagnose bestätigen kann. Einer *Diagnose* können mehrere *Prämissen* zugeordnet werden. Beispiel:

Diagnose 1 trifft dann zu wenn: **Prämisse 1**
 (Symptom 1 und Symptom 4 und Symptom 6)
 oder **Prämisse 2**
 (Symptom 3 und nicht Symptom 2)
 zutrifft.

Jedem der Objekte *Symptom*, *Komponente* und *Diagnose* kann eine *Erklärung* zugeordnet werden. *Erklärungen* erläutern dem Anwender den Sachverhalt der *Diagnose* zur Laufzeit.

6.2.2 Objektrelationen

Komponenten können nach zwei verschiedenen Ordnungsprinzipien einander zugeordnet werden. Unterschieden wird zwischen hierarchischer und heterarchischer [122] Zuordnung. Innerhalb eines Wissensbasismoduls kann das Ordnungsprinzip nicht geändert werden. Deshalb werden entsprechend die Wissensbasismodule als hierarchische oder heterarchische Wissensbasismodule bezeichnet. Bild 6.2 veranschaulicht die beiden Ordnungsprinzipien.

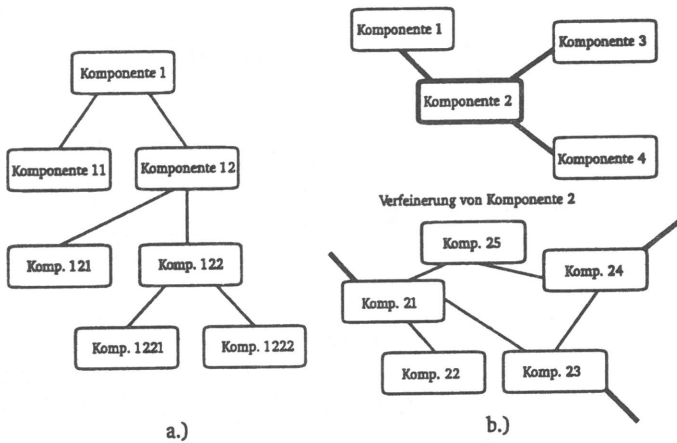


Bild 6.2 a.): Hierarchische Zuordnung von Komponenten
b.): Heterarchische Zuordnung von Komponenten

In einer hierarchischen organisierten Wissensbasis kann eine Komponente mehrere Nachfolgerkomponenten haben, jedoch nur eine Vorgängerkomponente. Bild 6.2.a zeigt die Struktur einer hierarchischen Wissensbasis. Die Nachfolgerkomponenten verfeinern die Vorgängerkomponente. Die erste Komponente eines Komponentenbaumes wird als Wurzelkomponente bezeichnet. Das hierarchische Ordnungsprinzip ist geeignet für die Beschreibung einzelner Netzkomponenten, wie z.B. PCs oder Workstations.

Für die Beschreibung einer Netzstruktur ist die heterarchische Beschreibungsweise besser geeignet. Bild 6.2.b zeigt die hierbei entstehende Modellstruktur. Komponenten werden in der heterarchischen Beschreibungsweise Beschreibungsebenen zugeordnet. Innerhalb einer Beschreibungsebene dürfen die Komponenten beliebig miteinander verknüpft werden, die ebenfalls netzartig miteinander verbunden sein dürfen. Während einer Diagnosesitzung werden nur die verdächtigen Komponenten verfeinert. Dadurch wird der aktuell bearbeitete Wissensbereich dem aktuellen Wissensstand angepaßt. Diese Vorgehensweise wird auch als dynamische Model-

lierung bezeichnet. Bild 6.3 zeigt in der Ebene 1 eine Komponente Netzwerk. Die Komponente Netzwerk wird in Ebene 2 durch die Komponenten $K1$, $K2$, $K3$ und $K4$ detailliert beschrieben. Im Lauf einer Diagnosesitzung werden $K2$ und $K4$ sehr verdächtig und daher verfeinert durch die entsprechenden Komponenten $K2i$ und $K4j$. Die Problemlösungskomponente arbeitet daher aktuell mit den Komponenten, die sich innerhalb der grau unterlegten Felder in Ebene 2 und Ebene 3 befinden.

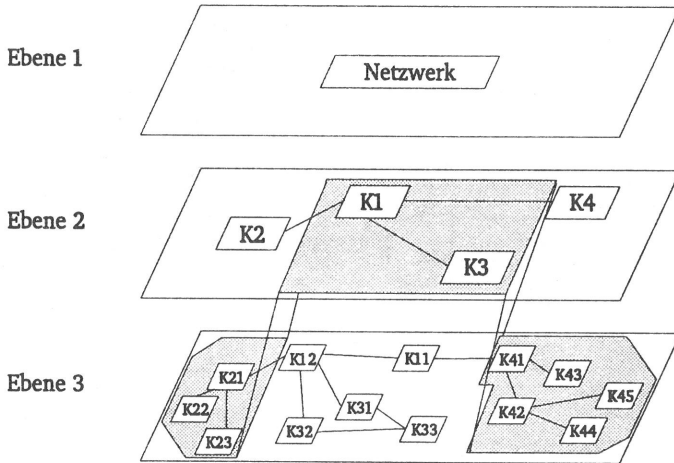


Bild 6.3 Beispiel für die dynamische Modellierung. Es werden nur die Komponenten verfeinert, die nach dem aktuellen fehlerspezifischen Wissen verdächtig werden.

Jeder Komponente können mehrere Diagnosen zugeordnet werden. Dabei wird zwischen Enddiagnosen und Diagnosen unterschieden. Nachfolger-Wissensbasen dürfen nur Komponenten zugeordnet werden, die nicht weiter verfeinert werden können. Kann eine Komponente nicht weiter verfeinert werden und ist ihr auch keine Nachfolger-Wissensbasis zugeordnet, dann dürfen ihr nur Enddiagnosen zugeordnet sein. Ist einer Komponente eine Nachfolger-Wissensbasis zugeordnet, dann muß der Komponente mindestens eine Diagnose zugeordnet sein.

Jeder Diagnose muß mindestens eine Prämisse zugeordnet sein. Die Bewertung einer Diagnose wird durch eine kompensatorische ODER-Verknüpfung der Bewertungen der ihr zugeordneten Prämissen bestimmt. Eine Prämisse besteht aus einem oder mehreren Symptomen. Die Bewertung einer Prämisse wird bestimmt durch eine kompensatorische UND-Verknüpfung der Bewertungen der ihr zugeordneten Symptome. Auf die Wissensbewertung wird in Abschnitt 6.2.3 näher eingegangen.

Jedem Symptom kann eine Regel zugeordnet werden. In einer Regel kann die Abhängigkeit eines Symptoms von anderen Symptomen ausgedrückt werden. Regeln werden angelehnt an die disjunktive Normalform der Booleschen Algebra angeben. Mit Regeln können unsinnige Fragen an den Anwender vermieden werden. Dadurch läßt sich die Akzeptanz des Diagnosesystems steigern. So kann z.B. dem Symptom "Es läßt sich keine Datenübertragung über das Netz durchführen" sofort intern eine zustimmende Bewertung zuordnen, wenn das Symptom "Der Abschlußwiderstand des Netzsegmentes ist defekt" bestätigt wird. Ein anderes Beispiel ist die Auswahl einer möglichen Farbe aus einer Farbpalette. Sobald die Frage nach einer Farbe zu stimmend beantwortet wurde, macht es keinen Sinn mehr, nach den anderen Farben zu fragen.

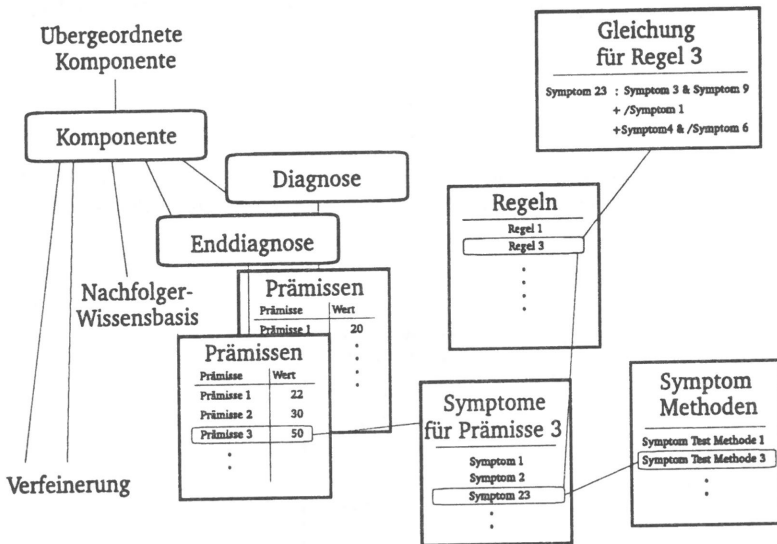


Bild 6.4 Darstellung der Objektrelationen. Außer der Komponentenzuordnung ist in beiden Organisationsprinzipien der Wissensdarstellung die Relation der Objekte untereinander identisch.

Jedem Symptom kann eine Methode zur automatischen Bewertung zugeordnet werden. Die Methode wird über die Monitoringschnittstelle an das Monitoringsystem weitergegeben. Alle für die Bewertung einer Komponente erforderlichen Methoden werden parallel gestartet. Durch den Aufruf einer Methode können andere Symptome mitbewertet werden. Damit sind Meßergebnisse, die oft als Nebeneffekt bei Messungen zur Bewertung für das aktuell zu bewertende Symptom anfallen, direkt mitverwertbar.

Jedem Objekt kann eine Erklärung zugeordnet werden. Die dem Objekt Erklärung zugeordnete Information wird als Systemaufruf an das Betriebssystem abgegeben. Damit lassen sich Hypertextsysteme [16, 31] oder Graphiksysteme für die Ausgabe von Informationen verwenden. In Bild 6.4 werden die Objektrelationen dargestellt.

6.2.3 Die Wissensbewertung

6.2.3.1 Darstellung der Wissensbewertung

Für die Problemlösung muß den Objekten "Diagnose", "Symptom", "Regel" und "Prämisse" eine Bewertung in Abhängigkeit des aktuellen Fehlerfalles zugeordnet werden. Da auch unscharfes Wissen verarbeitet werden soll, reicht eine binäre Darstellung der aktuellen Bewertung nicht aus. Daher wird für die Bewertung eine Bewertungsskala eingeführt. Für den Bewertungswert BW eines Objektes O gilt:

$$100 \geq BW(O) \geq -100 \quad (\text{Gl. 6.1})$$

Die Grenzen des Intervalles für den Bewertungswert $BW(O)$ wurden willkürlich festgelegt. Bild 6.5 zeigt die Abbildung linguistischer Bewertungen auf das Intervall des Bewertungswertes.

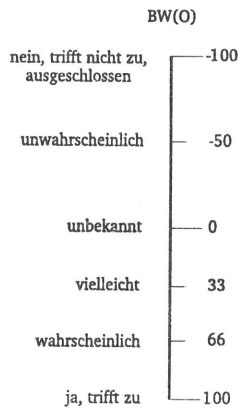


Bild 6.5 Zuordnung linguistischer Bewertungen zum Intervall des Bewertungswertes $BW(O)$

Für eine Menge von Bewertungswerten wurde eine Reihe von Verknüpfungen (Operatoren) definiert. Diese Verknüpfungen und deren Anwendungen sollen in den nächsten Abschnitten erläutert werden.

6.2.3.2 Operatoren für die Wissensbewertung

6.2.3.2.1 Die Negation

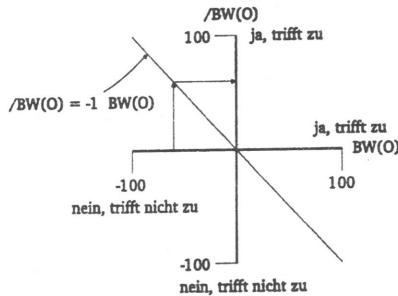


Bild 6.6 Veranschaulichung der Negation eines Bewertungswertes

Der Bewertungswert $BW(O)$ eines Objektes kann durch Anwendung der Negation in sein Gegenteil $/BW(O)$ überführt werden. Dazu wird der Bewertungswert mit -1 multipliziert. Das Zeichen $/$ vor einem Bewertungswert drückt dessen Negation aus. Bild 6.6 veranschaulicht die Negation.

$$/BW(O) = -1 \cdot BW(O) \quad (\text{Gl. 6.2})$$

Damit lassen sich aus zutreffenden Aussagen nicht zutreffende Aussagen ableiten. Ein Beispiel: Es soll die Farbe eines einfarbigen Autos bestimmt werden. Wird die Aussage $S1$: "Das Auto ist rot" mit dem Bewertungswert $BW(S1) = 90$ bewertet, so kann davon ausgegangen werden, daß das Auto nicht grün ist. Die Aussage $S2$: "Das Auto ist grün" trifft mit ziemlicher Sicherheit nicht zu. Es gilt:

$$BW(S2) = -1 \cdot BW(S1) = /BW(S1) = -90 \quad (\text{Gl. 6.2a})$$

6.2.3.2.2 Die UND-Verknüpfung

Die UND-Verknüpfung wird durch den Minimum-Operator realisiert. Der Minimum-Operator weist dem Bewertungswert $BW(E)$ eines Objektes E den minimalen Bewertungswert einer Menge $VK = \{BW(O_1), \dots, BW(O_k)\}$ (Verknüpfungsmenge) von Bewertungswerten zu. Die Menge VK enthält k Bewertungswerte.

Bild 6.7 veranschaulicht die UND-Verknüpfung. Der Minimum-Operator kann mit der UND-Verknüpfung der binären Logik verglichen werden. Da in diesem Anwendungsfall mit Wertebereichen gearbeitet wird, bezeichnet man die Operation, die durch den Minimum-Operator durchgeführt wird, auch als linguistische UND-Verknüpfung. Die Ausgangsgröße wird durch die niederwertigste Eingangsgröße bestimmt.

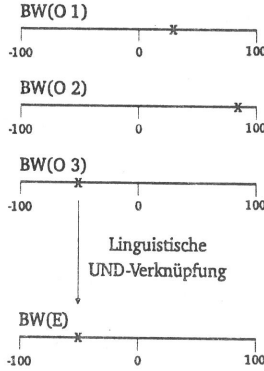


Bild 6.7 Die UND-Verknüpfung. Der Minimum-Operator wird auf die Bewertung der Objekte $O_i, i \in \{1,2,3\}$ angewandt. Das Ergebnis der Bewertung durch den Minimum-Operator wird der Bewertung des Objektes E zugewiesen.

$$BW(E) = \min[BW(O_1) \dots BW(O_k)] = \min[BW(O_i)]; \quad i = 1, 2, \dots, k \quad (\text{Gl. 6.3a})$$

Den einzelnen Objektbewertungen $BW(O_i)$ kann eine Gewichtung $GW(BW(O_i)) = GW(O_i)$ zugeordnet werden. $GW(O_i)$ stellt eine verkürzte Schreibweise für $GW(BW(O_i))$ dar. Die Gewichtung $GW(O_i)$ drückt die Relevanz aus, mit der ein Bewertungswert zu einer Verknüpfung beiträgt. Der Gewichtungswert wird durch eine natürliche Zahl aus dem Intervall $[1, 100]$ dargestellt. Dabei wird durch einen hohen Wert eine hohe, durch einen niedrigen Wert eine niedrige Relevanz ausgedrückt.

Um die Gewichtung $GW(O_i)$ der Bewertungswerte innerhalb einer UND-Verknüpfung zu berücksichtigen, wird die gefundene minimale Bewertung mit ihrer Gewichtung multipliziert und durch die maximale Gewichtung, die innerhalb der Verknüpfungsmenge VK auftritt, dividiert (Gl. 6.3b). Dadurch wird erreicht, daß ein minimaler Bewertungswert mit relativ niedriger Gewichtung abgeschwächt wird. Das Ergebnis der Verknüpfung wandert in diesem Fall immer gegen den Wert Null sowohl von der positiven als auch von der negativen Seite her. Minimale Bewertungswerte mit hoher Gewichtung werden kaum oder nicht abgeschwächt.

$$BW(E) = \frac{\min[BW(O_i)] \cdot \max[GW(O_i)]}{\max[GW(O_i)]}; \quad i = 1, 2, \dots, k \quad (\text{Gl. 6.3b})$$

6.2.3.2.3 Die ODER-Verknüpfung

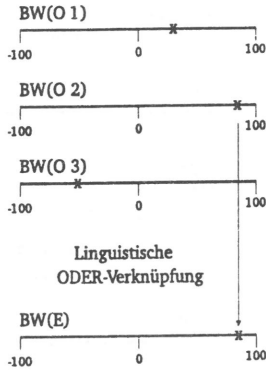


Bild 6.8 Die ODER-Verknüpfung. Der Maximum-Operator wird auf die Bewertung der Objekte $O_i, i \in \{1,2,3\}$ angewandt. Das Ergebnis der Bewertung durch den Maximum-Operator wird als Bewertung dem Objekt E zugewiesen.

Die ODER-Verknüpfung wird durch den Maximum-Operator realisiert. Der Maximum-Operator weist dem Bewertungswert $BW(E)$ eines Objektes E den maximalen Bewertungswert einer Menge $VK = \{BW(O_1) \dots BW(O_k)\}$ von Bewertungswerten zu. Bild 6.8 veranschaulicht diese Operation. Im Vergleich mit der binären Logik wird die Verknüpfung, die durch den Maximum-Operator realisiert wird, auch als linguistische ODER-Verknüpfung bezeichnet. Die höchstwertigste Eingangsgröße bestimmt die Ausgangsgröße.

$$BW(E) = \max[BW(O_1) \dots BW(O_k)] = \max[BW(O_i)]; \quad i = 1, 2, \dots, k \quad (\text{Gl. 6.4a})$$

Die Gewichtung $GW(O_i)$ der einzelnen Objektbewertungen $BW(O_i)$ kann analog der UND-Verknüpfung berücksichtigt werden (Gl. 6.4b).

$$BW(E) = \frac{\max[BW(O_i)] \cdot \max[GW(O_i)]}{\max[GW(O_i)]}; \quad i = 1, 2, \dots, k \quad (\text{Gl. 6.4b})$$

Auch in Gleichung 6.4b wird durch Multiplizieren mit der Gewichtung des maximalen Bewertungswertes und durch Dividieren durch den maximalen, in der Verknüpfungsmenge VK vorkommenden, Gewichtungswert sichergestellt, daß maximale Bewertungswerte mit geringer Gewichtung abgeschwächt werden.

6.2.3.2.4 Der Mittelwert-Operator

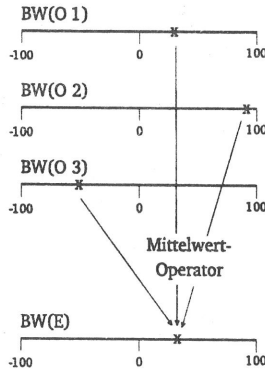


Bild 6.9 Der Mittelwert-Operator. Der Mittelwert-Operator wird auf die Bewertungen der Objekte $O_i, i \in \{1,2,3\}$ angewendet. Das Ergebnis der Bewertung wird der Bewertung des Objektes E zugewiesen.

Der Mittelwert-Operator weist dem Bewertungswert $BW(E)$ eines Objektes E den Mittelwert der Bewertungswerte einer Menge $VK = \{BW(O_1) \dots BW(O_k)\}$ von Bewertungswerten zu. Die Menge VK enthält k Bewertungswerte. Bild 6.9 veranschaulicht diese Operation.

$$BW(E) = \frac{1}{k} \sum_{i=1}^k BW(O_i) \quad (\text{Gl. 6.5})$$

Wird die Gewichtung $GW(O_i)$ der einzelnen Objektbewertungen $BW(O_i)$ berücksichtigt, dann gilt Gleichung 6.6 für die Bestimmung des Mittelwerts.

$$\begin{aligned}
 BW(E) &= \frac{\sum_{i=1}^k BW(O_i) \cdot GW(O_i)}{\frac{1}{k} \sum_{i=1}^k GW(O_i)} & (\text{Gl. 6.6}) \\
 &= \frac{\sum_{i=1}^k BW(O_i) \cdot GW(O_i)}{\sum_{i=1}^k GW(O_i)}
 \end{aligned}$$

6.2.3.2.5 Der kompensatorische UND-Operator

Die UND-Verknüpfung, die durch den Minimum-Operator realisiert wird, vernachlässigt sehr hohe Bewertungswerte der Verknüpfungsmenge, auf die sie angewandt wird. Diese "harte" Entscheidung entspricht nicht der menschlichen Vorgehensweise bei der Suche nach Diagnosen. Aus diesem Grund wird ein Parameter λ_{und} eingeführt, der die "harte" UND-Verknüpfung kompensiert. Der Parameter ist eine reelle Zahl mit folgendem Wertebereich:

$$0 \leq \lambda_{und} \leq 1 \quad (\text{Gl. 6.7})$$

Die Kompensation erfolgt durch die Addition des mit $(1 - \lambda_{und})$ gewichteten Mittelwertes der zur Verknüpfungsmenge VK gehörenden Bewertungswerte, zum mit λ_{und} gewichteten minimalen Bewertungswert (Gl. 6.8a).

$$BW(E) = \lambda_{und} \cdot \min[BW(O_i)] + (1 - \lambda_{und}) \frac{1}{k} \sum_{i=1}^k BW(O_i) \quad (\text{Gl. 6.8a})$$

Mit Berücksichtigung der Gewichtungswerte der Bewertungswerte erhält man Gleichung 6.8b.

$$BW(E) = \lambda_{und} \cdot \frac{\min[BW(O_i)] \cdot GW(\min[BW(O_i)])}{\max[GW(O_i)]} + (1 - \lambda_{und}) \frac{\sum_{i=1}^k BW(O_i) \cdot GW(O_i)}{\sum_{i=1}^k GW(O_i)} \quad (\text{Gl. 6.8b})$$

Für $\lambda_{und} = 1$ wird durch die kompensatorische UND-Verknüpfung eine UND-Verknüpfung realisiert; für $\lambda_{und} = 0$ eine Mittelwertbildung. Die Einstellung des Parameters λ_{und} muß empirisch ermittelt werden und ist abhängig davon, wie, pessimistisch oder optimistisch, sich der Anwender verhält (Gängiger Wert: $\lambda_{und} = 0,8$).

6.2.3.2.6 Der kompensatorische ODER-Operator

Analog zur kompensatorischen UND-Verknüpfung kann auch in die ODER-Verknüpfung ein Parameter λ_{oder} eingeführt werden. Der Wertebereich für λ_{oder} ist identisch mit dem Wertebereich von λ_{und} .

$$0 \leq \lambda_{oder} \leq 1 \quad (\text{Gl. 6.9})$$

Die Kompensation erfolgt hier durch die Addition des, mit $(1 - \lambda_{oder})$ gewichteten, Mittelwertes zum, mit λ_{oder} gewichteten, maximalen Bewertungswert der Verknüpfungsmenge VK (Gl. 6.10a).

$$BW(E) = \lambda_{oder} \cdot \max[BW(O_i)] + (1 - \lambda_{oder}) \cdot \frac{1}{k} \sum_{i=1}^k BW(O_i) \quad (Gl. 6.10a)$$

Mit Berücksichtigung der Gewichtung der Bewertungswerte ergibt sich Gleichung 6.10b.

$$BW(E) = \lambda_{oder} \cdot \frac{\max[BW(O_i)] \cdot GW(\max[BW(O_i)])}{\max[GW(O_i)]} + (1 - \lambda_{oder}) \cdot \frac{\sum_{i=1}^k BW(O_i) \cdot GW(O_i)}{\sum_{i=1}^k GW(O_i)} \quad (Gl. 6.10b)$$

Für $\lambda_{oder} = 1$ wird durch die kompensatorische ODER-Verknüpfung eine ODER-Verknüpfung realisiert; für $\lambda_{oder} = 0$ eine Mittelwertbildung. Die Einstellung des Parameters λ_{oder} muß, wie die Einstellung von λ_{undt} , empirisch ermittelt werden (auch hier ist ein gängiger Wert $\lambda_{oder} = 0,8$).

6.2.3.3 Die Bewertung von Regeln

Zur Darstellung der Beziehung von Symptomen untereinander werden Regeln verwendet. Regeln werden analog der disjunktiven Normalform der Booleschen Algebra angegeben. Zunächst wird die Bewertung der einzelnen UND-Verknüpfungsterme (Produktterme) durch Anwendung des MIN-Operators berechnet. Danach wird der Bewertungswert $BW(Regel)$ durch Anwendung des MAX-Operators auf die Bewertungswerte der zugehörigen UND-Verknüpfungen bestimmt. Der Bewertungswert einer Regel $BW(Regel)$ darf nur dann dem Bewertungswert eines Symptoms $BW(S)$ zugewiesen werden, wenn das Symptom durch die Regel sehr sicher bestätigt oder nicht bestätigt wird. Ansonsten bleibt das Symptom unbewertet und muß durch eine Methode bewertet werden (Resolutionsverfahren). Das Ergebnis einer Regel wird deshalb nur verwendet, wenn dessen Betrag einen bestimmten Schwellenwert TH_{Regel} übersteigt. Der Schwellenwert TH_{Regel} ist konfigurierbar (Default: 80).

$$BW(S) = \begin{cases} BW(Regel) & \text{wenn } |BW(Regel)| > TH_{Regel} \\ \text{unbewertet} & \text{sonst} \end{cases} \quad (Gl. 6.11)$$

Eine Regel wird nur ausgewertet, wenn die Symptome mindestens eines Produktterms bekannt sind.

Den Symptomen einer Regel kann ein Attribut zugeordnet werden, welches sicherstellt, daß das Symptom zur Bewertung der Regel nur verwendet wird, wenn es eindeutig positiv oder negativ bewertet wurde. Siehe hierzu auch Abschnitt 6.4.2.

6.2.3.4 Die Bewertung von Prämissen

Einer Diagnose können mehrere Prämissen zugeordnet werden. Der Bewertungswert der Prämissen $BW(Prämisse)$ wird durch die kompensatorische UND-Verknüpfung mit Berücksichtigung der Gewichtungen der Symptome bestimmt (Gl. 6.8b.). Auf die Gewichtung der Symptome innerhalb von Prämissen wird in Abschnitt 6.4.2. näher eingegangen. Der Bewertungswert der Diagnose $BW(Diagnose)$ wird durch die kompensatorische ODER-Verknüpfung aller Prämissenbewertungen anhand Gleichung 6.9a bestimmt. Eine Zuordnung von Gewichtungswerten zu den Prämissen ist nicht vorgesehen.

6.3 Modulare Wissensbasis

6.3.1 Aufbau der Wissensbasis

Das Wissen für die Diagnose eines lokalen Rechnernetzes mit seiner Vielzahl von unterschiedlichen Hardware- und Software-Komponenten kann nicht in einer einzigen Wissensbasis untergebracht werden. Eine solche Wissensbasis kann nicht konsistent gehalten werden [52]. Auch ist es äußerst schwierig, diese Wissensbasis auf andere Rechnernetze anzupassen.

Auch in der Realität ist das Wissen über große Gesamtsysteme auf mehrere Personen verteilt. In der Regel gibt es einen Systemexperten, der die Fehlersuche soweit durchführt, bis ein oder mehrere fehlerhafte Subsysteme isoliert werden können. Für die Fehlerdiagnose innerhalb der Subsysteme werden dann Bereichsexperten hinzugezogen. Bild 6.10 veranschaulicht die Aufgabenteilung zwischen System- und Bereichsexperten.

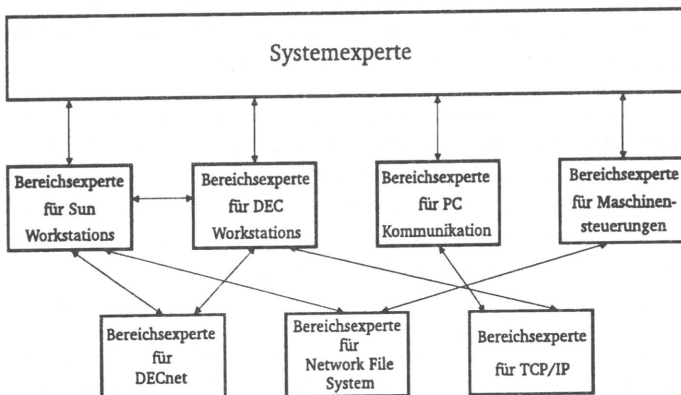


Bild 6.10

Problemlösung in großen Systemen durch System- und Bereichsexperten.

Die Vorgehensweise bei der Problemlösung ist iterativ. Der Systemmanager versucht, ausgehend von der Betrachtung des Gesamtsystems, das fehlerhafte Subsystem zu isolieren. Ist dies gelungen, gibt es zwei weitere Möglichkeiten:

1. Entfernen des fehlerhaften Subsystems und Ersetzen durch ein fehlerfreies Subsystem
2. Beauftragen eines Bereichsexperten für das fehlerhafte Subsystem, die Fehlersuche innerhalb des Subsystems fortzusetzen.

Dies kann sich über mehrere Ebenen fortsetzen. Der Bereichsexperte kann wiederum ein von ihm erkanntes defektes Subsystem ersetzen oder einen weiteren Bereichsexperten mit der Fehlersuche beauftragen.

Analog zu der in der Realität vollzogenen Aufteilung des Wissens besteht eine Wissensbasis aus einem oder mehreren Wissensbasismodulen. In einem Wissensbasismodul wird das Diagnosewissen für die Systemarchitektur beschrieben. Dieses Wissensbasismodul wird auch als Wurzelwissensbasis bezeichnet. Bei der Anwendung des Diagnosesystems für ein lokales Rechnernetz wird in der Wurzelwissensbasis das Diagnosewissen für die gesamte Netzstruktur einschließlich der am Rechnernetz angeschlossenen Geräte und deren Kommunikationsbeziehungen beschrieben. Mit der Wurzelwissensbasis ist daher die Fehlersuche bis zu einer defekten Netzkomponente möglich.

Für die Fehlersuche innerhalb der defekten Netzkomponenten können dann weitere Wissensbasismodule über das Objekt "Nachfolger-Wissensbasis" an die Wurzelwissensbasis angebunden werden. Allgemein können Wissensbasismodule in beliebiger Verschachtelung mit dem Objekt "Nachfolger-Wissensbasis" aneinander angebunden werden.

Die Wissensbasismodule sind unabhängig voneinander erstellbar. Damit wird gewährleistet, daß die Wissensbasismodule von den jeweiligen Experten für das betreffende Wissensgebiet erstellt werden können. Jedes Wissensbasismodul stellt eine Wissensbasis für ihren speziellen Wissensbereich dar und kann auch als Wissensbasis für diesen Bereich verwendet werden. Ein Beispiel: Ein Wissensbasismodul, das für die Fehlersuche in einem Personal Computer (PC) realisiert wurde, kann sowohl als Wissensbasis für die Reparatur eines PC verwendet werden als auch als Wissensbasismodul für die Fehlersuche in einem System, in dem der PC eingesetzt wird.

Die Modularisierung hat somit zur Folge, daß einzelne Wissensbasismodule mehrfach verwendet werden können. Das oben genannte Wissensbasismodul für PCs kann in allen Wissensbasen für die Fehlerdiagnose von Systemen verwendet werden, in denen dieser PC-Typ verwendet wird. Stehen für die Komponenten eines Rechnernetzes Wissensbasismodule zur Verfügung, so müssen für den Aufbau einer Wissensbasis für das Rechnernetz lediglich die Wurzelwissensbasis und die Verbindungen zwischen den einzelnen Wissensbasismodulen realisiert werden.

6.3.2 Zusammenbindung der Wissensbasismodule

Wissensbasismodule werden untereinander durch das Objekt "Nachfolger-Wissensbasis" miteinander verknüpft. Prinzipiell wird jedes Wissensbasismodul für sich bearbeitet. Wird ein neues Wissensbasismodul geladen, dann besteht jedoch die Möglichkeit bekanntes, vom aktuellen Fehlerfall abhängiges, Wissen für die Arbeit mit dem neuen Wissensbasismodul zu verwenden. Zu diesem Zweck wird die Agenda des Diagnosesystems unterteilt in eine globale Agenda und mehrere lokale Agenden. In den lokalen Agenden wird das fallspezifische Wissen der einzelnen Wissensbasismodule gespeichert. In der globalen Agenda das fehlerspezifische Wissen, das für zwei oder mehr Wissensbasismodule von Interesse ist.

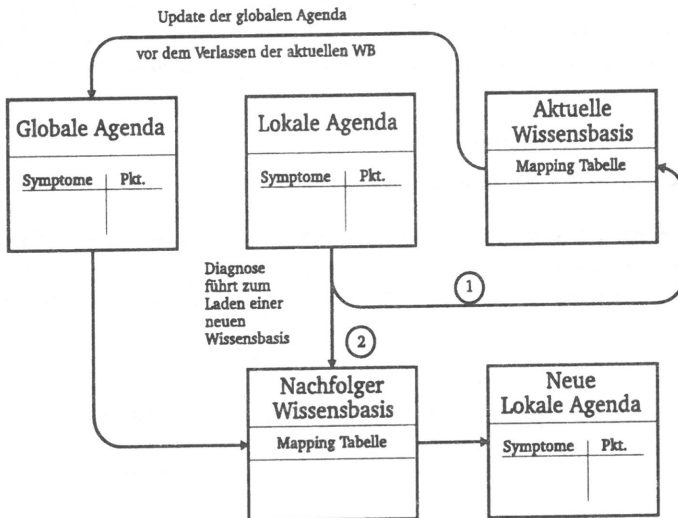


Bild 6.11

Verdachtsgenerierung beim Laden eines neuen Wissensbasismoduls. Zunächst wird die globale Agenda anhand der Mapping-Tabelle des aktuell bearbeiteten Wissensbasismoduls auf den neuesten Stand gebracht (1). Danach werden das neue Wissensbasismodul und eine neue lokale Agenda geöffnet. Anhand der Mapping-Tabelle des neuen Wissensbasismoduls und der globalen Agenda werden bekannte Symptome und deren Bewertungen in die neue lokale Agenda eingetragen (2). Die Problemlösung wird anhand des bekannten fall-spezifischen Wissens mit dem geladenen Wissensbasismodul fortgesetzt.

Wenn Wissen durch unterschiedliche Experten beschrieben wird, so muß davon ausgegangen werden, daß unterschiedliche Bezeichnungen für ein und denselben Sachverhalt verwendet werden. Ein Beispiel: Ein Experte verwendet für die Kollisionen, die beim gleichzeitigen Senden

von 2 Stationen über das Übertragungsmedium beim CSMA/CD Zugriffsverfahren entstehen, den Begriff "Kollisionen". Ein weiterer Experte verwendet die englische Bezeichnung "Collision". Das Diagnosesystem kann diese beiden Begriffe nicht ohne weiteres als identisch erkennen. Daher muß im Diagnosesystem eine Tabelle gehalten werden, welche die Zuordnung gleichbedeutender Begriffe zueinander ermöglicht.

Symptome sind die Objekte, deren Bewertungen beim Bearbeiten eines anderen Wissensbasismoduls von Bedeutung sein können. Daher wird in jeder Wissensbasis, in einer sogenannten Mapping-Tabelle, angegeben, welches lokale Symptom identisch mit einem globalen Symptom ist. Beim Laden eines neuen Wissensbasismoduls (Bild 6.11) wird dann zunächst anhand der Mapping-Tabelle des aktuellen Wissensbasismoduls die globale Agenda auf den neuesten Stand gebracht. Danach wird das neue Wissensbasismodul und eine neue lokale Agenda geöffnet. Anhand der Mapping-Tabelle des neu geöffneten Wissensbasismoduls und der globalen Agenda werden bekannte Symptome und deren Bewertung in die neue lokale Agenda eingetragen. Die Bearbeitung des neuen Wissensbasismoduls kann dann mit bekanntem Wissen gestartet werden.

6.4 Die Wissenserwerbskomponente

6.4.1 Architektur der Wissenserwerbskomponente

Der Wissenserwerb ist einer der Problempunkte beim Aufbau eines Diagnosesystems. Experten bestimmter Wissensgebiete sind häufig überlastet und stehen für die Dokumentation ihres Fachwissens nicht zur Verfügung. Häufig erhält daher ein sogenannter Wissensingenieur (Knowledge

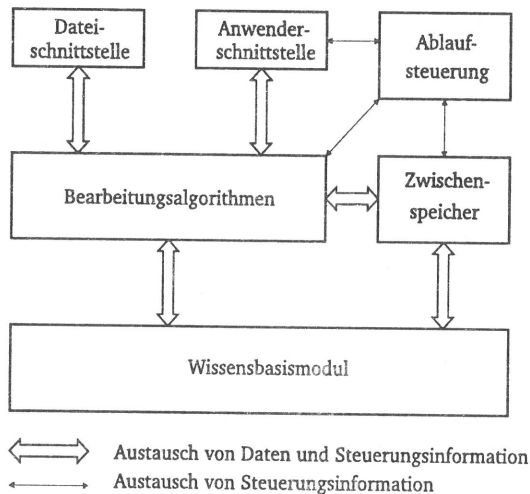


Bild 6.12 Architektur der Wissenserwerbskomponente

Engineer) die Aufgabe, die Experten eines Fachgebietes zu befragen und sie bei ihrer Tätigkeit zu beobachten, um dann ein Wissensbasismodul zu erstellen. Dabei wird häufig der Wissensingenieur selbst zum Experten.

Aufgabe der Wissenserwerbskomponente des Diagnosesystems ist es, den Wissensingenieur, oder den Experten selbst, bei der Erstellung von Wissensbasismodulen zu unterstützen. Bild 6.12 zeigt die Architektur der Wissenserwerbskomponente.

Die Wissenserwerbskomponente verfügt über 2 Schnittstellen für den Wissenserwerb, die Anwender- und die Dateischnittstelle. Die Anwenderschnittstelle wird zur Steuerung der Wissenserwerbskomponente und zur interaktiven Erfassung und Änderung von Wissen verwendet. Über die Dateischnittstelle können Wissensbeschreibungsdateien eingelesen und ausgegeben werden.

Kern der Wissenserwerbskomponente stellt die Ablaufsteuerung dar. Die Ablaufsteuerung bedient sich der Bearbeitungsalgorithmen zur Durchführung des Wissenserwerbs. Dazu wird der Inhalt der Wissensbeschreibungsdatei interpretiert. Die Wissensobjekte und deren Relationen zueinander werden im Zwischenspeicher abgelegt. Nach Erfassen abgeschlossener Teile der Wissensbeschreibungsdatei werden die entsprechenden Tabellen des Wissensbasismoduls angelegt und der Inhalt des Zwischenspeichers in das Wissensbasismodul übertragen.

Die Wissenserwerbskomponente kann in zwei unterschiedlichen Modi betrieben werden:

- Betrieb über die Dateischnittstelle
- Interaktiver Betrieb über die Anwenderschnittstelle

Beim Betrieb über die Dateischnittstelle wird aus einer Wissensbeschreibungsdatei ein Wissensbasismodul generiert. Ebenfalls kann aus einem Wissensbasismodul eine Wissensbeschreibungsdatei generiert werden. Der Inhalt von Wissensbasen kann im interaktiven Betrieb geändert werden.

Die Wissenserfassung sollte über die Dateischnittstelle durchgeführt werden, da die Wissensbeschreibungsdatei zugleich eine gute Dokumentation des erfaßten Wissens darstellt. Eine Wissensbeschreibungsdatei kann zudem mit Kommentaren versehen werden. Auf den Aufbau der Wissensbeschreibungsdatei wird im folgenden Kapitel eingegangen.

6.4.2 Syntax zur Wissensbeschreibung

Eine Wissensbeschreibungsdatei ist in verschiedene Blöcke untergliedert. Es werden folgende Blöcke unterschieden:

- **Header-Block**
In diesem Block wird der Namen des Wissensbasismoduls und dessen Ordnungsprinzip festgelegt (hierarchisch oder heterarchisch).

- **Vernetzungblock**
Definition der einzelnen Komponenten und deren Verbindungen untereinander.

- **Symptomblock**
In diesem Block werden die verwendeten Symptome und ihre Ermittlungsmethoden definiert.

- **Regelblock (optional)**
Mit Regeln wird der Zusammenhang von Symptomen untereinander beschrieben. Man bezeichnet die Abhängigkeit von Symptomen untereinander auch als Metawissen.

- **Frame-Block**
Im Frame-Block wird den Komponenten das Diagnosewissen zugeordnet. Jeder Komponente, die innerhalb des Vernetzungsblockes definiert wurde, muß mindestens eine Diagnose zugeordnet werden. Einer Diagnose wiederum muß mindestens eine Prämisse zugeordnet werden.

- **Mapping-Block (optional)**
Der Mapping-Block bildet den Abschluß einer Wissensbeschreibungsdatei. Er enthält Informationen darüber, welche lokal definierten Symptome auf global definierte Symptome abgebildet werden können. Siehe hierzu auch Abschnitt 6.3.2.

Die genannten Blöcke müssen in der angegebenen Reihenfolge in der Wissensbeschreibungsdatei erscheinen. Die mit "optional" gekennzeichneten Blöcke können entfallen. Ein Block wird durch sogenannte Steuerworte eingeleitet und durch weitere Steuerworte untergliedert. Die Steuerworte bestehen aus einem Einleitungszeichen "*" (Asterisk) und einer aus Großbuchstaben bestehenden Zeichenkette.

Eine Wissensbeschreibungsdatei wird immer durch den Header-Block eingeleitet. Der Header-Block hat folgenden Aufbau:

```
*HEADER (leitet den Header-Block ein)
  *WB_NAME: "Name des Wissensbasismoduls"
  *ORG: hierarchisch|heterarchisch
```

Das Steuerwort ***HEADER** leitet den Block ein. Nach dem Steuerwort ***WB_NAME** wird in Anführungszeichen der Name des Wissensbasismoduls angegeben. Durch das Steuerwort ***ORG** wird das Ordnungsprinzip des Wissensbasismoduls festgelegt. Es kann hier entweder "hierarchisch" oder "heterarchisch" angegeben werden.

Im Vernetzungsblock, der durch das Steuerwort ***VERNETZUNG** eingeleitet wird, werden die Komponenten für die Systembeschreibung definiert und deren Beziehung untereinander festgelegt. Die Bezeichnungen für Komponenten werden immer in Anführungszeichen angegeben.

Durch die erste Verwendung einer Bezeichnung für eine Komponente wird diese zugleich definiert.

Wird das hierarchische Ordnungsprinzip angewendet, dann wird nur noch das Steuerwort ***VERBINDUNG** benötigt. Eine Verbindung im hierarchischen Ordnungsprinzip bedeutet zugleich, daß die links stehende Komponente durch die rechts stehende Komponente verfeinert wird:

```
*HEADER
  *WB_NAME: "Name des Wissensbasismoduls"
  *ORG: hierarchisch
*VERNETZUNG (leitet den Vernetzungsblock ein)
*VERBINDUNG
  "Komponente" -- "Komponente 1", Häufigkeitswert der Verbindung (HW1)
  "Komponente" -- "Komponente 2", HW2
  "Komponente" -- "Komponente 3", HW3

  "Komponente 3" -- "Komponente 3_1", HWi
  "...

```

Durch die oben angegebene Vernetzung werden der Komponente "Komponente" die Komponenten "Komponente 1" bis "Komponente 3" zugeordnet. Die Komponente "Komponente 3" wird weiter unterteilt in die Komponente "Komponente 3_1". Die Verfeinerung von Komponenten wird im hierarchischen Ordnungsprinzip durch die Angabe der Verbindung der Komponenten implizit mitangegeben. Bild 6.13 stellt die, im Beispiel angegebene, hierarchische Ordnung dar.

Im hierarchischen Ordnungsprinzip wird den einzelnen Verbindungen ein *Häufigkeitswert (HW)* zugeordnet. HW ist eine natürliche Zahl des Intervalles [1, 100]. Ist eine Komponente mit Nachfolgerkomponenten verdächtig, dann wird zunächst die Nachfolgerkomponente untersucht, deren Verbindung der größte Häufigkeitswert zugeordnet ist. Haben mehrere Verbindungen zu Nachfolgerkomponenten denselben Häufigkeitswert, dann werden alle Nachfolgerkomponenten untersucht.

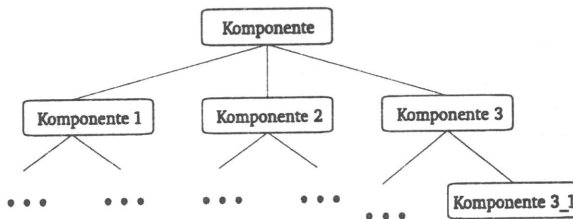


Bild 6.13

Beispiel für eine hierarchische Anordnung von Komponenten

Wird die heterarchische Beschreibungsweise angewendet, so muß die Verfeinerung von Komponenten explizit angegeben werden. Dazu wird das Steuerwort ***VERFEINERUNG** verwendet. Im folgenden Beispiel wird die Komponente "Komponente 1" verfeinert in die Komponenten "Komponente 1_1", "Komponente 1_2" und "Komponente 1_3". Die Komponente 2 wird verfeinert in die "Komponenten 2_1" und die "Komponente 2_2". Zwischen den Komponenten "Komponente 1" und "Komponente 2" besteht eine Verbindung. Diese Verbindung muß auch in den entsprechenden Komponentenverfeinerungen zu finden sein. In der Verfeinerung von "Komponente 1" hat die "Komponente 1_3" eine Verbindung zu "Komponente 2". In der Verfeinerung von "Komponente 2" hat die "Komponente 2_1" eine Verbindung zu "Komponente 1". Damit besteht eine Verbindung zwischen "Komponente 2_1" und der "Komponente 1_3" in der verfeinerten Beschreibungsebene. Bild 6.14 stellt die im Beispiel angegebene heterarchische Beschreibung dar.

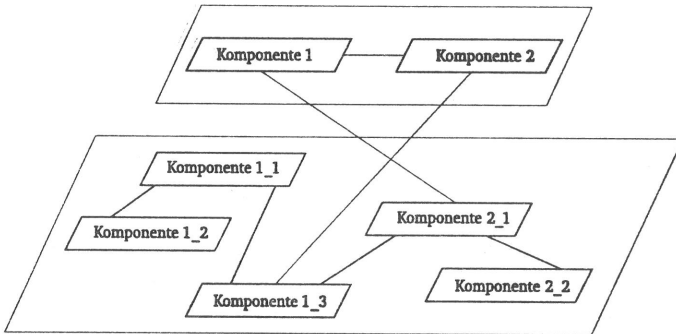


Bild 6.14 Graphische Darstellung des Beispiels für die heterarchische Beschreibungsweise.

```
*HEADER
  *WB_NAME: "Name des Wissensbasismoduls"
  *ORG: heterarchisch
*VERNETZUNG (leitet den Vernetzungsblock ein)
*VERFEINERUNG
  "Komponente 1"
  ("Komponente 1_1",
  "Komponente 1_2",
  "Komponente 1_3");
*VERNETZUNG
  "Komponente 1_1" -- "Komponente 1_2"
  "Komponente 1_1" -- "Komponente 1_3"
  "Komponente 1_3" -- "Komponente 2"
*VERFEINERUNG
  "Komponente 2"
  ("Komponente 2_1",
  "Komponente 2_2");
```



```
*VERNETZUNG
  "Komponente 2_1" -- "Komponente 1"
  "Komponente 2_1" -- "Komponente 2_2"
  "..."
```

Das Steuerwort ***SYMPTOM** leitet den Definitionsblock für Symptome ein. Optional können den Symptomen, eingeleitet durch das Steuerwort ***METH**, Methoden und, eingeleitet durch das Steuerwort ***INFO**, Erklärungen zugewiesen werden:

```
*SYMPTOM (leitet den Symptom-Block ein)
  *SYM "Text zu Symptom 1"
    *METH "Systemaufruf für Methode zu Symptom 1", timeout
    *INFO "Systemaufruf zu Ausgabe einer Erklärung zu Symptom 1"
  *SYM "Text zu Symptom 2"
    *METH "Systemaufruf für Methode zu Symptom 2", timeout
  *SYM ...
```

Wird einem Symptom keine Methode zugeordnet, dann wird der Anwender um die Bewertung gebeten (Default-Methode). Der nach dem Steuerwort ***METH** angegebene Systemaufruf wird mit dem zugeordneten *timeout* Wert an die Monitoringschnittstelle übergeben. Der übergebene *timeout* Wert dient zur Überwachung der Bearbeitungszeit der Methode. Liegt nach Ablauf des Zeitintervalls, das durch den Parameter in der Einheit Minuten angegeben wird, das Ergebnis der Methode nicht vor, so wird die Methode angehalten und das betreffende Symptom interaktiv bewertet. Auf die Monitoringschnittstelle wird in Abschnitt 6.5.5 näher eingegangen.

Wird vom Anwender eine Erklärung zum Symptom gewünscht, dann wird der unter ***INFO** angegebene Systemaufruf an das Betriebssystem übergeben. Damit lassen sich vorhandene Hilfstexte oder Graphiken in das System einbinden. Das Steuerwort ***INFO** kann auch Komponenten und Diagnosen zugeordnet werden.

Optional folgt nach dem Symptomblock der Regelblock. Dieser Block wird eingeleitet durch das Steuerwort ***REGEL**. Vor jeder Regel wird das Steuerwort ***SYMREG** angegeben. Der Regel wird eine Bezeichnung zugewiesen.

```
*REGEL (Einleitung des Regelblocks)
  *SYMREG "Regel für Symptom k"
    "Symptom k" : "Symptom 1" & "Symptom 2"
                  + "Symptom 3" & "Symptom 4"
                  + "Symptom 5" (1)
                  + "Symptom 6" (0)
                  + /"Symptom 8"
```

Im Beispiel wird die Bezeichnung *"Regel für Symptom k"* der Regel für *"Symptom k"* zugewiesen. UND-Verknüpfungen werden durch das Zeichen '&' dargestellt. Zunächst werden die UND-Verknüpfungen berechnet. Danach wird das Ergebnis der ODER-Verknüpfung bestimmt. Die ODER-Verknüpfung wird durch das Zeichen '+' dargestellt.

"Symptom 8" wird negiert in die Regel mit einbezogen. Die Kennzeichnung erfolgt durch voranstellen des Zeichens '/' vor die Bezeichnung des Symptoms. *"Symptom 5"* wird nur beachtet,

wenn dafür eine positive Bewertung vorliegt. Deshalb wurde das Symptom nachfolgend nach dessen Bezeichnung mit (1) gekennzeichnet. "Symptom 6" wird nur beachtet, wenn eine negative Bewertung des Symptoms vorliegt. Dies wird durch (0) gekennzeichnet. Die Kennzeichnung, ob ein Symptom nur bei vorliegender positiver oder negativer Bewertung zur Regelauswertung verwendet werden darf, kann auch der gesamten Regel zugewiesen werden. Dazu wird dem Symptom, das durch die Regel bewertet werden soll, die Kennzeichnung (1) oder (0) nachgestellt.

Es darf maximal eine Regel für ein Symptom angegeben werden. Dies stellt keine Einschränkung dar, da durch die Regelsyntax die disjunktive Normalform gebildet wird.

Durch das Steuerwort ***FRAME** wird der Frame-Block eingeleitet. Im Frame-Block werden den Komponenten Diagnosen, Enddiagnosen und evtl. Nachfolger-Wissensbasen zugeordnet. Den Diagnosen werden im Frame-Block Prämissen zugewiesen, die sich aus Symptomen zusammensetzen.

```
*FRAME (leitet den Frame-Block ein)
  *KOMP "Komponente k"
    *DIAG "Diagnose 1 für Komponente k"
      *PREMISE
        *SYM "Symptom 1", GW(Symptom 1)
        *SYM "Symptom n", GW(Symptom n)
      *PREMISE
        *SYM / "Symptom 3", GW(Symptom 3)
        *SYM / "Symptom 5", GW(Symptom 5)
        *SYM "Symptom m", GW(Symptom m)
      *ENDDIAG "Enddiagnose 1 für Komponente k"
        *PREMISE
          *SYM / "Symptom 7", GW(Symptom 7)
          *SYM "Symptom i", GW(Symptom i)
      *NACHF_WB "Name der Nachfolger-Wissensbasis"
```

Durch das Steuerwort ***KOMP** wird die Beschreibung einer Komponente eingeleitet. Die Bezeichnung der Komponente folgt dem Steuerwort nach. ***DIAG** weist der Komponente eine Diagnose zu, deren Bezeichnung diesem Steuerwort nachfolgt. Mit diesem Steuerwort wird die entsprechende Diagnose zugleich definiert.

Das Steuerwort ***PREMISE** weist der zuvor definierten Diagnose eine Prämisse zu, deren Symptome nachfolgend aufgeführt werden. Jedes Symptom einer Prämisse wird durch das Steuerwort ***SYM** eingeleitet. Dem Steuerwort folgt die Bezeichnung des Symptoms. Vor der Bezeichnung eines Symptoms ist die Negation '/' zugelassen. Wird ein Symptom durch '/' negiert, so wird zur Bestimmung des Bewertungswertes der Prämisse der negierte Bewertungswert des betreffenden Symptoms verwendet.

Den Symptomen wird eine Gewichtung $GW(\text{Symptom})$ zugeordnet. Die Gewichtung ist nur innerhalb der aktuellen Prämisse gültig. Die Gewichtung ist ein Maß dafür, wie stark das Symptom bei der Bewertung der Prämisse berücksichtigt werden soll (vergleiche hierzu Abschnitt 6.2.3.2.2 und 6.2.3.2.3). Die Gewichtung wird durch eine natürliche Zahl aus dem

Intervall [1, 100] dargestellt. Ein hoher Wert gibt eine hohe, ein niedriger Wert eine niedrige Gewichtung.

Die Verknüpfung der Bewertungswerte der Symptome einer Prämisse erfolgt mit dem kompensatorischen UND-Operator (Siehe auch Abschnitt 6.2.3.2.5).

Einer Diagnose können mehrere Prämissen zugeordnet werden. Der Bewertungswert einer Diagnose wird durch die kompensatorische ODER-Verknüpfung der Bewertungswerte der Prämissen bestimmt.

Durch das Steuerwort ***ENDDIAG** wird eine Enddiagnose definiert. Wird eine Enddiagnose ausreichend bewertet, dann wird diese als Fehlerursache dem Anwender angeboten. Nach Bestätigung durch den Anwender wird die Fehlersuche abgeschlossen. Die Diagnosesitzung wird auch dann abgeschlossen, wenn die Komponente, der die Enddiagnose zugeordnet ist, hätte weiter verfeinert werden können. Komponenten, denen keine Verfeinerung und keine Nachfolger-Wissensbasis zugeordnet wird dürfen nur Enddiagnosen zugeordnet werden.

Einer Enddiagnose werden, wie einer Diagnose, Prämissen und Symptome unter Verwendung derselben Steuerworte zugeordnet.

***NACHF_WB** ordnet der Komponente ein Nachfolgerwissensbasismodul zu. Dem Steuerwort folgt die Bezeichnung des Nachfolgerwissensbasismoduls. Das Nachfolgerwissensbasismodul wird geladen, wenn eine Diagnose der Komponente bestätigt wird.

Der optionale Mapping-Block wird durch das Schlüsselwort ***MAPPING** eingeleitet. Im Mapping Block werden Symptome des Wissensbasismoduls globalen Symptomen zugeordnet. Dadurch ist es möglich aktuell gewonnenes Wissen an weitere Wissensbasismodule zu übergeben (Siehe hierzu auch Abschnitt 6.3.2).

```
... *MAPPING
    "Symptom 1" -- "Globales Symptom n"
    "Symptom 2" -- "Globales Symptom m"
    ...
```

6.5 Die Problemlösungskomponente

6.5.1 Architektur der Problemlösungskomponente

Die Architektur der Problemlösungskomponente des Diagnosesystems zeigt Bild 6.15.

In der Agenda der Problemlösungskomponente wird das aktuelle, fehlerabhängige Wissen gespeichert. Dazu gehören die verdächtigten Komponenten, deren Diagnosen, die bekannten Symptome und deren aktuelle Bewertungen. Mit den Agendaoperationen kann der Inhalt der Agenda verändert werden.

Aufgerufen werden die Agendaoperationen von der Ablaufsteuerung. Jede Agendaoperation prüft zunächst den Inhalt der Agenda und ändert danach, anhand der modularen Wissensbasis

und Informationen, die über die Schnittstellen der Problemlösungskomponente erfragt werden, deren Inhalt. Jede Agendaoperation gibt nach ihrer Abarbeitung eine Statusmeldung an die Ablaufsteuerung zurück. Anhand der Statusmeldung, dem aktuellen Zustand der Agenda und berechneter interner Parameter zur Problemlösungssteuerung wird von der Ablaufsteuerung entschieden, welche Agendaoperation anschließend durchgeführt wird.

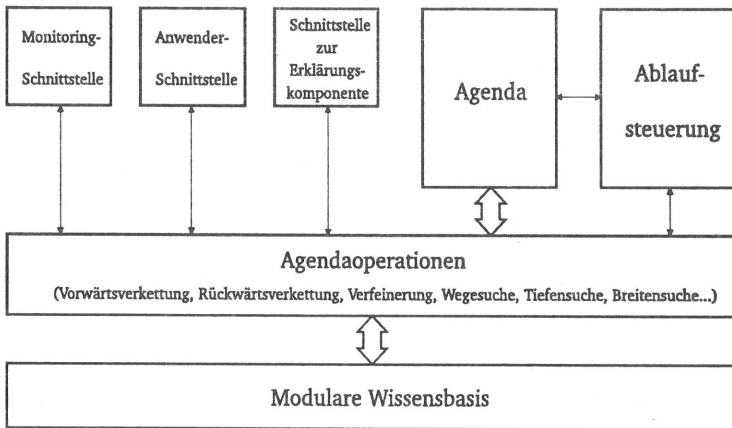


Bild 6.15 Architektur der Problemlösungskomponente des Diagnosesystems.

Über die Schnittstelle zur Erklärungskomponente werden Informationen über die durchgeführten Problemlösungsalgorithmen und deren Ergebnisse an die Erklärungskomponente weitergegeben.

6.5.2 Agendaoperationen

6.5.2.1 Einteilung der Agendaoperationen

Im folgenden werden die zur Problemlösung verwendeten Agendaoperation vorgestellt. Die Agendaoperationen werden untergliedert in allgemeine Agendaoperationen und Agendaoperationen für hierarchische oder heterarchische Wissensbasismodule. Zu den allgemeinen Agendaoperationen zählen alle Operationen, die zur Verwaltung der Agenda benötigt werden. Operationen, die sowohl auf hierarchische als auch auf heterarchische Wissensbasismodule angewendet werden können, zählen ebenso zu den allgemeinen Agendaoperationen. Die Agendaoperationen, die nur auf hierarchische Wissensbasismodule angewandt werden können, werden als hierarchische Agendaoperationen bezeichnet. Agendaoperationen für heterarchische Wissensbasismodule werden unter dem Begriff heterarchische Agendaoperationen zusammengefasst.

Eine Untermenge der Menge aller Agendaoperationen bilden die Agendaoperationen, die als Operationsergebnis eine oder mehrere Komponenten in die Agenda eintragen. Diese Operationen werden auch als "Komponenten eintragende Agendaoperationen" bezeichnet. Der Aufruf einer "Komponenten eintragenden Agendaoperation" markiert einen Zustand einer Teilablaufsteuerung.

6.5.2.2 Allgemeine Agendaoperationen

Komponentenverdächtigung

Die Komponentenverdächtigung trägt zu Beginn einer Problemlösung verdächtige Komponenten in die Agenda ein. Der Anwender kann angeben welche Komponenten eines Systems ihm besonders verdächtig erscheinen. Werden keine Komponenten vom Anwender ausgewählt, so wird die Wurzelkomponente des geladenen Wissensbasismoduls verwendet.

Vorwärtsverkettung

Die Vorwärtsverkettung (Forward Chaining) geht von bekanntem Wissen aus, um neue Verdächtigungen zu generieren. Anhand der in der Agenda eingetragenen und bewerteten Symptome, deren Bewertungswert $BW(Symptom) > TH_{Symptom}$ ist, trägt die Agendaoperation "Vorwärtsverkettung" alle Komponenten ein, bei denen bekannte Symptome einen positiven Beitrag zur Bewertung einer Diagnose liefern. Dabei werden evtl. Negationen in Prämissen mitberücksichtigt.

Rückwärtsverkettung

Die Rückwärtsverkettung (Backward Chaining) geht von einer unbewerteten (und damit verdächtigen) Komponente aus und versucht diese zu bestätigen. Dazu werden alle der Komponente zugeordneten Diagnosen und Symptome in die Agenda eingetragen. Danach wird die Agendaoperation "Symptombewertung" aufgerufen.

Symptombewertung

Diese Agendaoperation entnimmt der Agenda alle unbewerteten Symptome und führt deren Bewertung durch. Dabei werden, in Abhängigkeit des Betriebsmodus, 3 Fälle unterschieden:

1. Die unbekanntenen Symptome werden alle vom Anwender bewertet.
2. Ist einem unbewerteten Symptom eine Methode zugeordnet, dann wird der Anwender gefragt, ob die Methode durchgeführt werden soll oder ob das Symptom interaktiv bewertet werden soll. Symptome, denen keine Methode zugeordnet ist, müssen vom Anwender interaktiv bewertet werden (Default-Methode).
3. Unbewertete Symptome, denen eine Methode zugeordnet ist, werden durch Methoden bewertet. Alle anderen Symptome werden interaktiv bewertet.

Für die automatische Bewertung von Symptomen werden die Methoden und deren Timeout-Parameter der Monitoringschnittstelle übergeben. Die interaktive Bewertung wird über die Anwender-Schnittstelle durchgeführt. Bei der Durchführung der Agendaoperation "Symptombewertung" werden, sofern vorhanden, Regeln für Symptome beachtet. Bild 6.16 zeigt ein vereinfachtes Flußdiagramm für die Agendaoperation "Symptombewertung" nach Fall 1.

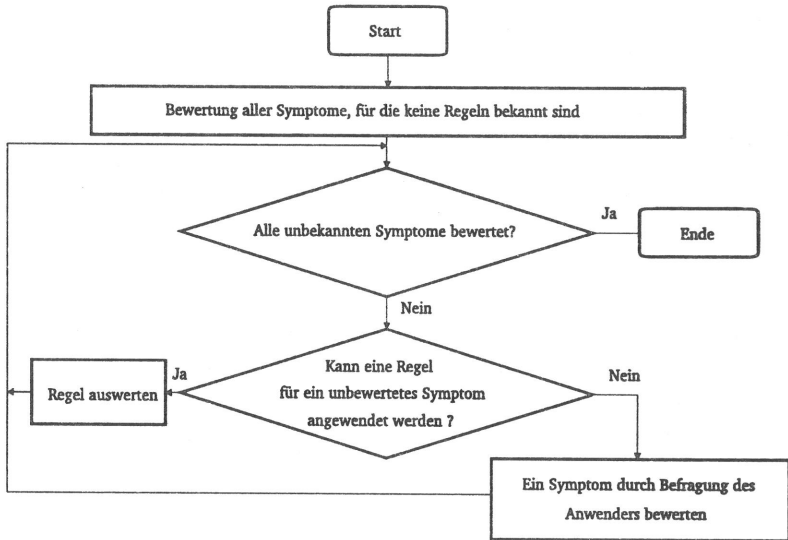


Bild 6.16 Flußdiagramm für die interaktive Symptombewertung unter Berücksichtigung von Regeln.

Diagnosebewertung

Wurde neues fehlerabhängiges Wissen in die Agenda eingetragen, so müssen die Diagnosen neu bewertet werden. Dies wird durch die Agendaoperation "Diagnosebewertung" durchgeführt. Die Agendaoperation "Diagnosebewertung" ist eng an die Agendaoperation "Symptombewertung" geknüpft.

Weitere allgemeine Agendaoperationen

Für die Durchführung einer Problemlösung sind weitere Agendaoperationen notwendig. Hierzu zählen:

- Operationen, um Wissensbasismodule zu öffnen und zu schließen
- Operationen für die Ergebnisausgabe

6.5.2.3 Agendaoperationen für hierarchische Wissensbasismodule

Tiefensuche

Die Agendaoperation "Tiefensuche" (Depth First) trägt in die Agenda die Nachfolgerkomponente der verdächtigsten Komponente ein, die noch nicht eingetragen ist und die den größten Häufigkeitswert *HW* besitzt. Existieren mehrere gleichverdächtige Komponenten, so wird die Operation auf die Komponente angewendet, die am weitesten verfeinert vorliegt. Existieren mehrere Nachfolgerkomponenten mit demselben Häufigkeitswert, dann werden alle diese Komponenten eingetragen.

Breitensuche

Die Agendaoperation "Breitensuche" (Breadth First) trägt die noch nicht eingetragene Komponente derselben Verfeinerungsebene in die Agenda ein, die den größten Häufigkeitswert *HW* besitzt. Sind mehrere Komponenten gleichverdächtig, wird die Operation mit der am weitesten verfeinerten Komponente durchgeführt. Existieren mehrere Nachfolgerkomponenten mit demselben Häufigkeitswert, dann werden alle diese Komponenten eingetragen.

Baumergänzung

Die Agendaoperation "Baumergänzung" bestimmt innerhalb der Agenda alle Komponenten, deren Vorgängerkomponenten noch nicht in die Agenda eingetragen wurden. Die fehlenden Vorgängerkomponenten werden eingetragen.

6.5.2.4 Agendaoperationen für heterarchische Wissensbasismodule

Verfeinerung

Die Agendaoperation "Verfeinerung" bestimmt die verdächtigste Komponente in der Agenda für die eine Verfeinerung im aktuellen Wissensbasismodul existiert. Die Verfeinerung der verdächtigsten Komponente wird in die Agenda eingetragen und für die weitere Problemlösung verwendet. Die verfeinerte Oberkomponente wird entsprechend gekennzeichnet.

Modellbildung

Die Agendaoperation "Modellbildung" vervollständigt das Modell mit dem gearbeitet wird. Dazu werden innerhalb der Agenda alle Komponenten bestimmt deren Oberkomponenten noch nicht in die Agenda eingetragen wurden. Die Oberkomponenten und deren Beziehungen zueinander werden in die Agenda eingetragen und als verfeinert gekennzeichnet.

Wegesuche

Die Agendaoperation "Wegesuche" bestimmt die Komponenten, die zwei verdächtige Komponenten miteinander verbinden und trägt diese in die Agenda ein.

6.5.3 Parameter zur Problemlösungssteuerung

Der Schwellenwert für die Bewertung einer Diagnose $TH_{Diagnose}$ gibt an, ab welcher Höhe einer Bewertung einer Diagnose die Diagnose akzeptiert werden kann. Der Wert von $TH_{Diagnose}$ ist vom Anwender konfigurierbar.

Enddiagnosen, welche die Bewertung $BW(Enddiagnose) = 100$ erhalten, werden von der Ablaufsteuerung als Problemlösung ohne Anwenderbefragung akzeptiert. Werden innerhalb eines Wissensbasismoduls sowohl alle Enddiagnosen als auch alle Diagnosen mit -100 (trifft nicht zu, ausgeschlossen) bewertet, dann wird die Problemlösung innerhalb des Wissensbasismoduls abgebrochen.

Während einer Problemlösung wird nach jeder Bewertung von Diagnosen die aktuelle maximale Diagnosebewertung $TH_{DynDiag}$ aller in der aktuellen Agenda eingetragenen i Diagnosen bestimmt. Für den Zustand n ergibt sich $TH_{DynDiag}$ aus:

$$TH_{DynDiag}^n = \max[BW(Diagnose_i)]^n \quad (\text{Gl. 6.12})$$

Die Differenz $\Delta_{DynDiag}$:

$$\Delta_{DynDiag}^{n+1} = TH_{DynDiag}^{n+1} - TH_{DynDiag}^n \quad (\text{Gl. 6.13})$$

stellt ein Maß dafür dar, ob eine Verbesserung der Problemlösung im Zustand $n+1$ gegenüber dem Zustand n erzielt wurde. Eine Verbesserung wurde erzielt, wenn $\Delta_{DynDiag} > 0$ ist. Ist $\Delta_{DynDiag} \leq 0$, dann wurde keine Verbesserung der Problemlösung erzielt und es muß evtl. eine andere Problemlösungsstrategie gewählt werden.

Der Schwellenwert TH_{Regel} (siehe hierzu Abschnitt 6.2.3.2) bestimmt, ab welchem Betrag einer Bewertung einer Regel, die Regel für die Bewertung eines Symptoms verwendet werden darf. TH_{Regel} ist durch den Anwender konfigurierbar.

Für die Verwendung von bewerteten Symptomen in der weiteren Problemlösung (speziell bei der Vorwärtsverkettung) wird die dynamische Schwelle $TH_{Symptom}$ eingeführt. $TH_{Symptom}$ wird zu Beginn einer Problemlösung auf den Wert 75 gesetzt. $TH_{Symptom}$ ist an $TH_{DynDiag}$ gebunden. Wenn $TH_{Symptom} > TH_{DynDiag}$ ist, dann wird $TH_{Symptom}$ herabgesetzt. Die Herabsetzung erfolgt jeweils in Stufen um 25 bis zu $TH_{Symptom} = 0$.

Für jede "Komponenten eintragende Agendaoperation" existiert eine Erfolgsvariable $EV(<Name\ der\ Agendaoperation>)$. Diese Variable wird auf den Wert "1" gesetzt, wenn durch irgendeine Agendaoperation eine Komponente in die Agenda eingetragen wurde. Die Erfolgsvariable einer Agendaoperation wird zurückgesetzt, wenn durch die betreffende Agendaoperation keine Komponente in die Agenda eingetragen werden konnte. Agendaoperationen,

deren Erfolgsvariable auf "0" gesetzt ist, werden durch die betreffende Teilablaufsteuerung nicht mehr aufgerufen.

Die Parameter der kompensatorischen Und- und Oder-Verknüpfung λ_{und} und λ_{oder} (Vergleiche hierzu Abschnitt 6.2.3.2.5 und 6.2.3.2.6), die einen Einfluß auf die Bewertung von Prämissen und Diagnosen haben, sind ebenfalls durch den Anwender konfigurierbar.

6.5.4 Die Ablaufsteuerung

6.5.4.1 Anforderungen an die Ablaufsteuerung

Die Ablaufsteuerung entscheidet über den Aufruf der Agendaoperationen. Ziel ist es, die Problemlösung schnell zu finden. Dabei soll die Anzahl der notwendigen Messungen und Anwenderbefragungen gering sein.

Die Problemlösung muß sicher gefunden werden, auch wenn zu Beginn einer Problemlösung der falsche Lösungsweg eingeschlagen wurde. Voraussetzung hierfür ist eine korrekt aufgebaute Wissensbasis. Allgemein gilt der Grundsatz: Was nicht in einer Wissensbasis festgehalten ist, kann auch nicht durch die Problemlösungskomponente gefunden werden.

Bild 6.17 zeigt den Aufbau der Ablaufsteuerung. Die Ablaufsteuerung ist weiter untergliedert in die zentrale Ablaufsteuerung und die speziellen Teilablaufsteuerungen für hierarchische und heterarchische Wissensbasismodule.

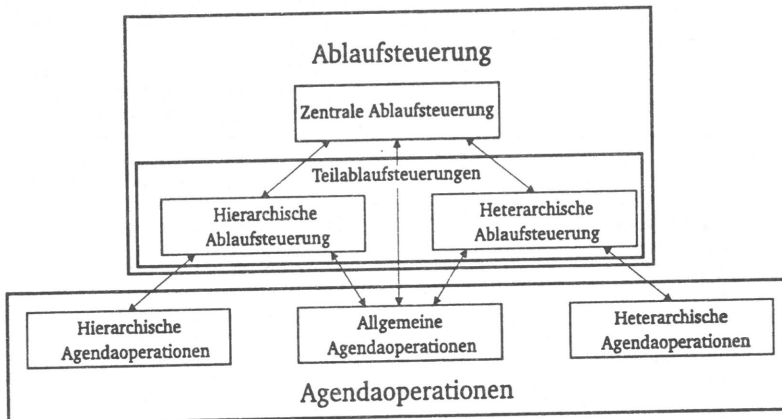


Bild 6.17 Aufbau der Ablaufsteuerung und Zuordnung der Agendaoperationen

Die zentrale Ablaufsteuerung und die beiden Teilablaufsteuerungen für hierarchische und heterarchische Wissensbasismodule wurden als Zustandsautomaten realisiert.

6.5.4.2 Die zentrale Ablaufsteuerung

Die zentrale Ablaufsteuerung steuert anhand der Teilablaufsteuerungen und der allgemeinen Agendaoperationen die Problemlösung. Nach dem Start der Ablaufsteuerung wird vom Anwender das gewünschte Wissensbasismodul erfragt. Das ausgewählte Wissensbasismodul wird geöffnet und es wird eine Komponentenverdächtigung durchgeführt. Die Komponentenverdächtigung trägt die vom Anwender verdächtigten Komponenten in die Agenda ein. Wird keine Komponente verdächtigt, dann wird die Wurzelkomponente des Wissensbasismoduls in die Agenda eingetragen. Die Ablaufkontrolle für die Problemlösung wird, entsprechend der Organisation des Wissensbasismoduls, an eine Teilablaufsteuerung übergeben.

Meldung der Teilablaufsteuerungen	Anzahl der geöffneten Wissensbasismodule	Aktion der zentralen Ablaufsteuerung
Alle Diagnosen und Enddiagnosen des untersuchten Wissensbasismoduls haben die Bewertung -100 (trifft nicht zu, ausgeschlossen).	1	Ausgabe einer Meldung an den Anwender, daß alle Diagnosen ausgeschlossen sind.
	>1	Rückkehr zum Vorgänger-Wissensbasismodul
Es wurde eine Enddiagnose gefunden mit einem Bewertungswert für den gilt: $BW(\text{Enddiagnose}) > TH_{\text{Diagnose}}$	- (don't care)	Dem Anwender wird die Enddiagnose angeboten. Es werden 2 Fälle unterschieden: <u>Fall 1:</u> Der Anwender akzeptiert die gestellte Diagnose. Die Problemlösung wird abgeschlossen und das Ergebnis ausgegeben. <u>Fall 2:</u> Der Anwender akzeptiert die angebotene Problemlösung nicht. Dann wird mit der Problemlösung fortgefahren.
Alle Komponenten des aktuellen Wissensbasismoduls wurden untersucht, keine Enddiagnose wurde höher als TH_{Diagnose} bewertet.	1	Die maximal verdächtigsten Enddiagnosen/ Diagnosen dem Anwender anbieten.
	>1	Die maximal verdächtigsten Enddiagnosen/ Diagnosen dem Anwender anbieten. Bei Nichtakzeptanz das geöffnete Wissensbasismodul schließen und mit dem Vorgänger-Wissensbasismodul weiterarbeiten.
Eine Diagnose mit zugeordnetem Nachfolger- Wissensbasismodul wurde gefunden. Die Bewertung der Diagnose ist höher als TH_{Diagnose} .	-	Zugeordnetes Nachfolger-Wissensbasismodul öffnen und die Problemlösung fortsetzen.

Tabelle 6.1 Aktionen der zentralen Ablaufsteuerung auf Meldungen der hierarchischen und der heterarchischen Teilablaufsteuerungen.

Die Ablaufsteuerungen für die entsprechenden Wissensbasismodule führen die Problemlösung solange durch, bis ein Abbruchkriterium erkannt wird und die Ablaufkontrolle wieder an die zentrale Ablaufsteuerung übergeben wird. Tabelle 6.1 faßt die Meldungen der Teilablaufsteuerungen an die zentrale Ablaufsteuerung zusammen und gibt die Aktionen an, die von der zentralen Ablaufsteuerung aufgrund der Meldungen durchgeführt werden. Dabei wird die Anzahl der geöffneten Wissensbasismodule beachtet.

6.5.4.3 Die Teilablaufsteuerungen

6.5.4.3.1 Gemeinsamkeiten beider Teilablaufsteuerungen

Die Teilablaufsteuerungen wurden, wie die zentrale Ablaufsteuerung, als Zustandsautomaten realisiert. Eine spezielle Klasse von Zuständen wird durch die "Komponenten eintragenden Agendaoperationen" gebildet. Die Zustände werden entsprechend den Agendaoperationen bezeichnet. Bild 6.18 stellt die Aktionen innerhalb eines Zustandes zusammen. Können durch die "Komponenten eintragende Agendaoperation" Komponenten in die Agenda eingetragen werden, dann werden nacheinander die Agendaoperationen Rückwärtsverkettung, Symptombewertung und Diagnosebewertung durchgeführt.

Zustandsbezeichnung: "Name der Komponenten eintragenden Agendaoperation"	
<Komponenten eintragende Agendaoperation>	
Konnten Komponenten in die Agenda eingetragen werden?	
nein	ja
<ul style="list-style-type: none"> - Zugehörige Erfolgsvariable auf "0" setzen 	<ul style="list-style-type: none"> - Zugehörige Erfolgsvariable auf "1" setzen - Rückwärtsverkettung - Symptombewertung - Diagnosebewertung
Bestimmen des Folgezustandes	

Bild 6.18 Struktur der Zustände der Teilablaufsteuerungen, in denen "Komponenten eintragende Agendaoperationen" durchgeführt werden.

Die "Komponenten eintragenden Zustände" für hierarchische Wissensbasismodule sind:

- Tiefensuche,
- Breitensuche,
- Baumergänzung und

- Vorwärtsverkettung.

Die "Komponenten eintragenden Zustände" für die heterarchischen Wissensbasismodule sind:

- Verfeinerung,
- Wegesuche,
- Modellbildung und
- Vorwärtsverkettung.

Aus dem aktuellen Stand der Parameter und den Statusmeldungen, der innerhalb des Zustandes durchgeführten Agendaoperationen wird die nachfolgende Agendaoperation bestimmt.

Treffen die in Tabelle 6.2 angegebenen Bedingungen zu, dann wird die Ablaufkontrolle aus allen "Komponenten eintragenden Zuständen" sowohl der hierarchischen als auch der heterarchischen Ablaufsteuerung an die zentrale Ablaufsteuerung zurückgegeben.

Aus dem Agendainhalt und den Problemlösungsparametern abgeleitete Bedingung	Nächster Zustand
Eine Diagnose, der ein Wissensbasismodul zugeordnet ist, wurde gleich oder höher als $TH_{Diagnose}$ bewertet.	Rückkehr zur zentralen Ablaufsteuerung (Öffnen des Nachfolger-Wissensbasismoduls).
Eine Enddiagnose wurde mit 100 (trifft zu) bewertet.	Rückkehr zur zentralen Ablaufsteuerung (Ausgabe des Ergebnisses).
Alle in die Agenda eingetragenen Diagnosen und Enddiagnosen sind mit -100 (trifft nicht zu, ausgeschlossen) bewertet.	Rückkehr zur zentralen Ablaufsteuerung (Siehe Tabelle 6.1).
Eine Enddiagnose wurde gleich oder höher als $TH_{Diagnose}$ bewertet.	Rückkehr zur zentralen Ablaufsteuerung (Gefundene Enddiagnose dem Anwender anbieten. Falls die angebotenen Enddiagnose nicht akzeptiert wird, wird das aktuelle Wissensbasismodul weiter untersucht).

Tabelle 6.2 Zusammenstellung aller Bedingungen, für welche die Ablaufkontrolle aus allen "Komponenten eintragenden Zuständen" an die zentrale Ablaufsteuerung zurückgegeben wird. Weitere spezielle Bedingungen werden bei der Beschreibung der einzelnen Zustände angegeben.

In beiden Teilablaufsteuerungen werden zusätzlich die Zustände:

- Start hierarchisch / Start heterarchisch,
- Agenda Prüfung hierarchisch / Agenda Prüfung heterarchisch und
- Abbruch Prüfung

definiert.

Die Zustände "Start" hierarchisch/heterarchisch und "Agenda Prüfung" hierarchisch/heterarchisch sind spezifisch für die jeweilige Teilablaufsteuerung. Die genannten Zustände werden in den betreffenden Teilabschnitten beschrieben. Der Zustand "Abbruch Prüfung" ist für beide Teilablaufsteuerungen identisch. Die jeweilige Teilablaufsteuerung geht in diesen Zustand, wenn keine "Komponenten eintragende Agendaoperation" mehr durchgeführt werden kann, wenn also alle Erfolgsvariablen $EV(<Name\ der\ Agendaoperation>)$ (Vergleiche Abschnitt 6.5.3) auf "0" gesetzt sind. Tabelle 6.3 gibt anhand der aktuellen Einstellung der Agenda und der Problemlösungsparameter die nachfolgenden Zustände für den Zustand "Abbruch Prüfung" an.

Aus dem Agendainhalt und den Problemlösungsparametern abgeleitete Bedingung	Nächster Zustand
$TH_{Symptom} = 0$ und keine "Komponenten eintragende Agendaoperation" kann noch durchgeführt werden (Die Erfolgsvariablen aller "Komponenten eintragenden Agendaoperationen" sind auf "0" gesetzt). Das aktuelle Wissensbasismodul wurde vollständig durchsucht.	Rückkehr zur zentralen Ablaufsteuerung.
$TH_{Symptom} > 0$. Verringern der Syptomschwelle $TH_{Symptom}$ um 25.	Vorwärtsverkettung

Tabelle 6.3 Zusammenstellung der Zustandsübergangsbedingungen für den Zustand "Abbruch Prüfung"

6.5.4.3.2 Die hierarchische Ablaufsteuerung

Die hierarchische Ablaufsteuerung kontrolliert die Problemlösung innerhalb hierarchisch organisierter Wissensbasismodule. Bild 6.19 zeigt das Zustandsdiagramm der hierarchischen Ablaufsteuerung.

Nach dem Öffnen eines Wissensbasismoduls mit hierarchischer Organisation wird die Ablaufkontrolle von der zentralen Ablaufsteuerung an die hierarchische Ablaufsteuerung übergeben. Diese befindet sich dann im Zustand "Start hierarchisch". Tabelle 6.4 gibt die Zustände und die zugehörigen Bedingungen an, in die vom Zustand "Start hierarchisch" gewechselt werden kann.

Als Zustandsübergangsbedingungen für die "Komponenten eintragenden Zustände" gelten zunächst die in Tabelle 6.2 angegebenen Bedingungen. Auf die zusätzlichen Zustandsübergangsbedingungen einzelner Zustände wird im folgenden eingegangen.

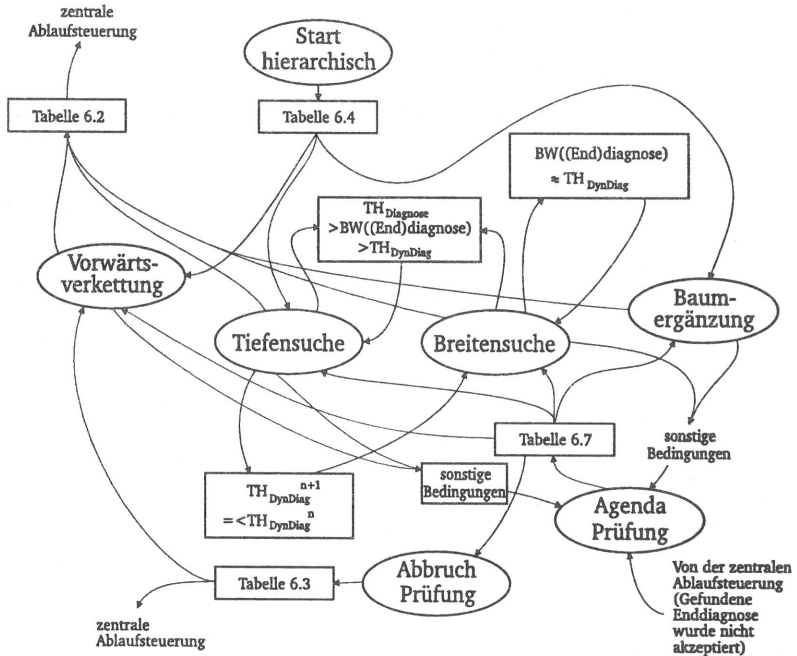


Bild 6.19 Zustandsdiagramm der hierarchischen Ablaufsteuerung.

Aus dem Agendainhalt und den Problemlösungsparametern abgeleitete Bedingung	Nächster Zustand
Mindestens ein Symptom und nur eine Komponente sind in der Agenda eingetragen (Ein Nachfolger-Wissensbasismodul wurde geladen, lokale Symptome wurden durch die globale Agenda bewertet).	Vorwärtsverkettung
In die Agenda ist nur eine Komponente (Wurzelkomponente) eingetragen.	Tiefensuche
Alle anderen Bedingungen	Baumergänzung

Tabelle 6.4 Zustandsübergangstabelle für den Zustand "Start hierarchisch".

Solange im Zustand "Tiefensuche" eine Diagnose oder Enddiagnose gefunden wird, deren Bewertungswert $BW((End)diagnose)$ steigt, also größer als $TH_{DynDiag}$ ist, wird wieder in den

Zustand "Tiefensuche" gegangen. Die relative Zunahme der Bewertungswerte läßt die Vermutung zu, daß sich die Problemlösung auf das richtige Ziel zubewegt. In Tabelle 6.5 werden die zusätzlichen Zustandsübergangsbedingungen für den Zustand "Tiefensuche" nochmals dargestellt.

Aus dem Agendainhalt und den Problemlösungsparametern abgeleitete Bedingung	Nächster Zustand
Siehe Tabelle 6.2	
Es wurde eine Komponente eingetragen mit einer Diagnose, deren Bewertung höher ist als die Bewertung aller bisherigen Diagnosen. $TH_{Diagnose} > BW(Diagnose) > TH_{DynDiag}$	Tiefensuche
Es wird eine Komponente eingetragen, deren Diagnosen sind aber geringer bewertet als alle bisherigen Diagnosen. $TH_{DynDiag}$ bleibt unverändert	Breitensuche

Tabelle 6.5 Zusätzliche Zustandsübergangsbedingungen für den Zustand "Tiefensuche".

Aus dem Agendainhalt und den Problemlösungsparametern abgeleitete Bedingung	Nächster Zustand
Siehe Tabelle 6.2	
Es wird eine Diagnose gefunden mit einer höheren Bewertung als alle bisherigen Diagnosen. $TH_{Diagnose} > BW(Diagnose) > TH_{DynDiag}$	Tiefensuche
Es wird eine Diagnose mit einer Bewertung gefunden, die etwa dem Wert von $TH_{DynDiag}$ entspricht. $BW(Diagnose) \approx TH_{DynDiag}$	Breitensuche

Tabelle 6.6 Zusätzliche Zustandsübergangsbedingungen für den Zustand "Breitensuche".

Aus dem Zustand "Breitensuche" wird in den Zustand "Tiefensuche" übergegangen, wenn eine Komponente mit einer höher bewerteten Diagnose gefunden wurde. Wird nur eine Diagnose gefunden, die einen Bewertungswert in derselben Größenordnung aufweist wie der aktuelle Wert von $TH_{DynDiag}$ vor der letzten Breitensuche, dann wird wieder in den Zustand "Breitensuche" gegangen. Tabelle 6.6 faßt die zusätzlichen Zustandsübergangsbedingungen zusammen.

Für alle nicht angegebenen Zustandsübergangsbedingungen wird aus allen "Komponenten eintragenden Zuständen" in den Zustand "Agenda Prüfung" der hierarchischen Ablaufsteuerung übergegangen. In diesem Zustand wird anhand des Agendainhaltes und der aktuellen Parameter für die Problemlösung entschieden, in welchen Zustand gewechselt wird. Tabelle 6.7 stellt die Zustandsübergangsbedingungen für den Zustand "Agenda Prüfung" zusammen.

Aus dem Agendainhalt und den Problemlösungsparametern abgeleitete Bedingung	Nächster Zustand
Vorhergehender Zustand: Vorwärtsverkettung	
Erfolgreich durchgeführte Vorwärtsverkettung. $EV(\text{Vorwärtsverkettung}) = 1$	Baumergänzung
Nicht erfolgreich durchgeführte Vorwärtsverkettung $EV(\text{Vorwärtsverkettung}) = 0$; Tiefensuche ist möglich $EV(\text{Tiefensuche}) = 1$.	Tiefensuche
Nicht erfolgreich durchgeführte Vorwärtsverkettung $EV(\text{Vorwärtsverkettung}) = 0$; Tiefensuche ist nicht möglich $EV(\text{Tiefensuche}) = 0$; Breitensuche ist möglich $EV(\text{Breitensuche}) = 1$.	Breitensuche
Nicht erfolgreich durchgeführte Vorwärtsverkettung $EV(\text{Vorwärtsverkettung}) = 0$; Tiefensuche ist nicht möglich $EV(\text{Tiefensuche}) = 0$; Breitensuche ist nicht möglich $EV(\text{Breitensuche}) = 0$; Baumergänzung ist möglich $EV(\text{Baumergänzung}) = 1$.	Baumergänzung
Keine "Komponenten eintragende Agendaoperation" ist möglich.	Abbruch Prüfung (Tabelle 6.3)
Vorhergehender Zustand: Tiefensuche	
Nicht erfolgreich durchgeführte Tiefensuche $EV(\text{Tiefensuche}) = 0$; Breitensuche ist möglich $EV(\text{Breitensuche}) = 1$.	Breitensuche
Nicht erfolgreich durchgeführte Tiefensuche $EV(\text{Tiefensuche}) = 0$; Breitensuche ist nicht möglich $EV(\text{Breitensuche}) = 0$; Vorwärtsverkettung ist möglich $EV(\text{Vorwärtsverkettung}) = 1$.	Vorwärtsverkettung

Nicht erfolgreich durchgeführte Tiefensuche $EV(Tiefensuche) = 0$; Breitensuche ist nicht möglich $EV(Breitensuche) = 0$; Vorwärtsverkettung ist nicht möglich $EV(Vorwärtsverkettung) = 0$; Baumergänzung ist möglich $EV(Baumergänzung) = 1$.	Baumergänzung
Keine "Komponenten eintragende Agendaoperation" ist möglich.	Abbruch Prüfung (Tabelle 6.3)
Vorhergehender Zustand: Breitensuche	
Nicht erfolgreich durchgeführte Breitensuche $EV(Breitensuche) = 0$; Tiefensuche ist möglich $EV(Tiefensuche) = 1$.	Tiefensuche
Nicht erfolgreich durchgeführte Breitensuche $EV(Breitensuche) = 0$; Tiefensuche ist nicht möglich $EV(Tiefensuche) = 0$; Vorwärtsverkettung ist möglich $EV(Vorwärtsverkettung) = 1$.	Vorwärtsverkettung
Nicht erfolgreich durchgeführte Breitensuche $EV(Breitensuche) = 0$; Tiefensuche ist nicht möglich $EV(Tiefensuche) = 0$; Vorwärtsverkettung ist nicht möglich $EV(Vorwärtsverkettung) = 0$; Baumergänzung ist möglich $EV(Baumergänzung) = 1$.	Baumergänzung
Keine "Komponenten eintragende Agendaoperation" ist möglich.	Abbruch Prüfung (Tabelle 6.3)
Vorhergehender Zustand: Baumergänzung	
Baumergänzung wurde durchgeführt, Tiefensuche ist möglich $EV(Tiefensuche) = 1$.	Tiefensuche
Baumergänzung wurde durchgeführt, Tiefensuche ist nicht möglich $EV(Tiefensuche) = 0$; Breitensuche ist möglich $EV(Breitensuche) = 1$.	Breitensuche
Baumergänzung wurde durchgeführt, Tiefensuche ist nicht möglich $EV(Tiefensuche) = 0$; Breitensuche ist nicht möglich $EV(Breitensuche) = 0$; Vorwärtsverkettung ist möglich $EV(Vorwärtsverkettung) = 1$.	Vorwärtsverkettung

Keine "Komponenten eintragende Agendaoperation" ist möglich.	Abbruch Prüfung (Tabelle 6.3)
--	-------------------------------

Tabelle 6.7 Zustandsübergangsbedingungen für den Zustand "Agenda Prüfung" der hierarchischen Ablaufsteuerung.

6.5.4.3.3 Die heterarchische Ablaufsteuerung

Die heterarchische Ablaufsteuerung kontrolliert die Problemlösung innerhalb heterarchisch organisierter Wissensbasen. Bild 6.20 zeigt das Zustandsdiagramm der heterarchischen Ablaufsteuerung.

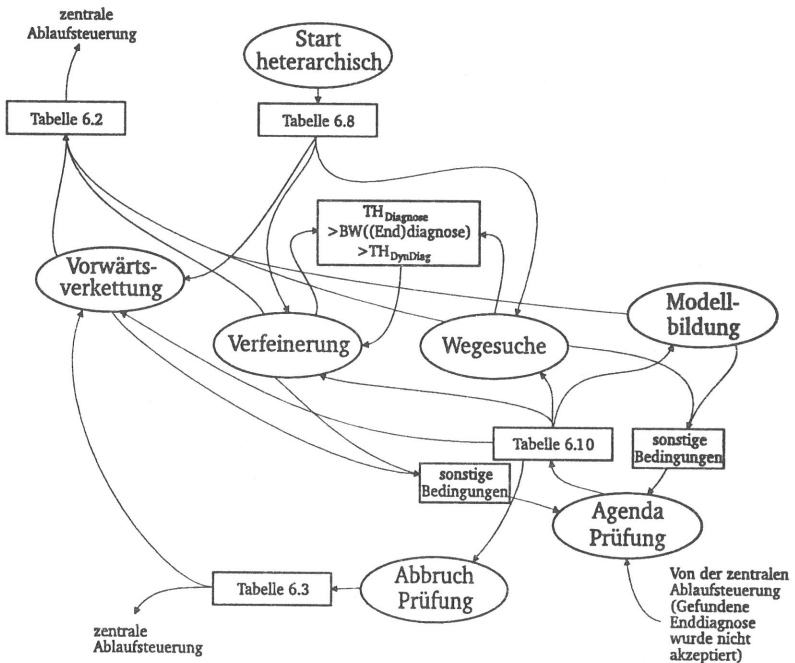


Bild 6.20 Zustandsdiagramm der heterarchischen Ablaufsteuerung.

Wurde durch die zentrale Ablaufsteuerung ein heterarchisches Wissensbasismodul geöffnet und die Ablaufkontrolle an die heterarchische Ablaufsteuerung übergeben, dann befindet sich diese zunächst im Zustand "Start heterarchisch". Abhängig vom Inhalt der Agenda wird entschieden,

in welchen Zustand der Ablaufsteuerung übergegangen wird. Tabelle 6.8 stellt die Bedingungen für die Zustandsübergänge zusammen.

Aus dem Agendainhalt und den Problemlösungsparametern abgeleitete Bedingung	Nächster Zustand
Mindestens ein Symptom und nur eine Komponente sind in der Agenda eingetragen (Ein Nachfolger-Wissensbasismodul wurde geladen, lokale Symptome wurden durch die globale Agenda bewertet).	Vorwärtsverkettung
In die Agenda ist nur eine Komponente (Wurzelkomponente) eingetragen.	Verfeinerung
Alle anderen Bedingungen	Wegesuche

Tabelle 6.8 Zustandsübergangstabelle für den Zustand "Start heterarchisch".

Für alle "Komponenten eintragenden Zustände" gelten zunächst die in Tabelle 6.2 angegebenen Zustandsübergangsbedingungen. Auf spezielle zusätzliche Bedingungen einzelner Zustände wird im folgenden eingegangen.

Solange im Zustand "Verfeinerung" neue Komponenten eingetragen werden, denen eine Diagnose oder Enddiagnose mit höherem Bewertungswert als der bisher ermittelte höchste Bewertungswert einer Diagnose $TH_{DynDiag}$ ist, wird wieder in den Zustand "Verfeinerung" gegangen. Die Zunahme von $TH_{DynDiag}$ läßt eine Untersuchung des aktuell geladenen Wissensbasismoduls in die richtige Richtung vermuten.

Aus dem Agendainhalt und den Problemlösungsparametern abgeleitete Bedingung	Nächster Zustand
Siehe Tabelle 6.2	
Es wurde eine Komponente eingetragen der eine Diagnose oder Enddiagnose zugeordnet ist mit einer im Vergleich höheren Bewertung $TH_{Diagnose} > BW(Diagnose) > TH_{DynDiag}$	Verfeinerung

Tabelle 6.9 Zusätzliche Zustandsübergangsbedingungen für die Zustände "Verfeinerung" und "Wegesuche".

Tritt dieselbe Bedingung, die im Zustand "Verfeinerung" zu einer weiteren Verfeinerung führt, im Zustand "Wegesuche" auf, dann wird ebenfalls in den Zustand "Verfeinerung" gegangen.

Tabelle 6.9 stellt die zusätzliche Zustandsübergangsbedingung für die Zustände "Verfeinerung" und "Wegesuche" dar.

Neben den angegebenen Bedingungen sind weitere Zustandsübergangsbedingungen möglich. Treten bisher nicht angegebene Bedingungen auf, dann wird in den Zustand "Agenda Prüfung" der heterarchischen Ablaufsteuerung gewechselt. In diesem Zustand wird anhand der aktuellen Werte der Problemlösungsparameter und des Agendainhalts entschieden, in welchen Folgezustand gewechselt wird. Tabelle 6.10 faßt die Zustandsübergangsbedingungen zusammen.

Aus dem Agendainhalt und den Problemlösungsparametern abgeleitete Bedingung	Nächster Zustand
Vorhergehender Zustand: Vorwärtsverkettung	
Verfeinerung ist möglich $EV(\text{Verfeinerung}) = 1$.	Verfeinerung
Verfeinerung ist nicht möglich $EV(\text{Verfeinerung}) = 0$; Wegesuche ist möglich $EV(\text{Wegesuche}) = 1$.	Wegesuche
Verfeinerung ist nicht möglich $EV(\text{Verfeinerung}) = 0$; Wegesuche ist nicht möglich $EV(\text{Wegesuche}) = 0$; Modellbildung ist möglich $EV(\text{Modellbildung}) = 1$.	Modellbildung
Keine "Komponenten eintragende Agendaoperation" ist möglich.	Abbruch Prüfung (Tabelle 6.3)
Vorhergehender Zustand: Verfeinerung	
Vorwärtsverkettung ist möglich $EV(\text{Vorwärtsverkettung}) = 0$;	Vorwärtsverkettung
Vorwärtsverkettung ist nicht möglich $EV(\text{Vorwärtsverkettung}) = 0$; Wegesuche ist möglich $EV(\text{Wegesuche}) = 1$.	Wegesuche
Vorwärtsverkettung ist nicht möglich $EV(\text{Vorwärtsverkettung}) = 0$; Wegesuche ist nicht möglich $EV(\text{Wegesuche}) = 0$; Modellbildung ist möglich $EV(\text{Modellbildung}) = 1$.	Modellbildung
Keine "Komponenten eintragende Agendaoperation" ist möglich.	Abbruch Prüfung (Tabelle 6.3)
Vorhergehender Zustand: Wegesuche	

Verfeinerung ist möglich $EV(\text{Tiefensuche}) = 1$.	Verfeinerung
Verfeinerung ist nicht möglich $EV(\text{Verfeinerung}) = 0$; Vorwärtsverkettung ist möglich $EV(\text{Vorwärtsverkettung}) = 1$.	Vorwärtsverkettung
Verfeinerung ist nicht möglich $EV(\text{Verfeinerung}) = 0$; Vorwärtsverkettung ist nicht möglich $EV(\text{Vorwärtsverkettung}) = 0$; Modellbildung ist möglich $EV(\text{Modellbildung}) = 1$.	Modellbildung
Keine "Komponenten eintragende Agendaoperation" ist möglich.	Abbruch Prüfung (Tabelle 6.3)
Vorhergehender Zustand: Modellbildung	
Verfeinerung ist möglich $EV(\text{Verfeinerung}) = 1$.	Verfeinerung
Verfeinerung ist nicht möglich $EV(\text{Verfeinerung}) = 0$; Wegesuche ist möglich $EV(\text{Wegesuche}) = 1$.	Wegesuche
Verfeinerung ist nicht möglich $EV(\text{Verfeinerung}) = 0$; Wegesuche ist nicht möglich $EV(\text{Wegesuche}) = 0$; Vorwärtsverkettung ist möglich $EV(\text{Vorwärtsverkettung}) = 1$.	Vorwärtsverkettung
Keine "Komponenten eintragende Agendaoperation" ist möglich.	Abbruch Prüfung (Tabelle 6.3)

Tabelle 6.10 Zustandsübergangsbedingungen für den Zustand "Agenda Prüfung" der heterarchischen Ablaufsteuerung.

6.5.5 Die Monitoringschnittstelle

Fehlerabhängiges Wissen wird in der Problemlösungskomponente durch die Bewertung von Symptomen erfaßt. Am einfachsten kann dies durch Befragen des Anwenders geschehen (Default-Methode). Komplexe Messungen und Tests müssen in diesem Fall durch den Anwender selbst durchgeführt, deren Ergebnisse interpretiert und der Problemlösungskomponente mitgeteilt werden. Ebenfalls vorgesehen ist die Zuordnung von Methoden zur automatischen Erfassung fehlerabhängigen Wissens zu Symptomen (Vergleiche Abschnitt 6.2.2). Explizit Symptomen zugeordnete Methoden sind z.B. Meß- oder Testprogramme. Durch die Verwendung von Methoden kann die Problemlösung um ein Vielfaches beschleunigt werden. Der Anwender wird entlastet.

Die Monitoringschnittstelle des Diagnosesystems bietet die Möglichkeit, fallspezifisches Wissen automatisch zu erfassen. Es handelt sich dabei um eine allgemeine Schnittstelle, die sich nicht nur an den Anforderungen des Monitoringsystems orientiert. Die Monitoringschnittstelle soll die Verwendung jedes fernsteuerbaren Gerätes und jeder vorhandenen Testsoftware ermöglichen.

Die Bewertung von Symptomen erfolgt durch die Agendaoperation "Symptombewertung". Allgemein kann angenommen werden, daß nach jeder "Komponenten eintragenden Agendaoperation" mit positivem Verlauf und anschließender Rückwärtsverketzung sich mindestens ein unbewertetes Symptom in der Agenda befindet (Vergleiche Abschnitt 6.4.2). Arbeitet die Problemlösungskomponente mit der Einstellung "Symptombewertung über Methoden", dann wird innerhalb der Agendaoperation "Symptombewertung" zunächst ermittelt, welche Symptome durch Methoden bewertet werden können. Danach wird, falls gewünscht, der Anwender gefragt, welche der bekannten Methoden durchgeführt werden sollen. Für die durchzuführenden Methoden wird eine Tabelle angelegt, in der jeder Methode eine Identifikationsnummer (Methoden-ID) zugeordnet wird. Diese Tabelle wird in Dateiform anderen Prozessen zugänglich gemacht. Für die durchzuführenden Methoden wird der maximale Timeout *MT* bestimmt. Danach werden die Methoden als parallele Prozesse gestartet. Dazu werden die Methodenbezeichnungen in Form von Systemaufrufen an das Betriebssystem übergeben. Bild 6.21 veranschaulicht in einem Flußdiagrammausschnitt der Agendaoperation "Symptombewertung" die Durchführung von Methoden.

Das Ergebnis einer Methode wird der Problemlösungskomponente in einer Datei mit der Bezeichnung <Methoden-ID>.dat zur Verfügung gestellt. Die Bewertung der Symptome wird darin als Text zeilenweise in folgender Form angegeben:

```
"Text zum Symptom x" $$<Bewertung x>
"Text zum Symptom y" $$<Bewertung y>
"Text zum Symptom n" $$<Bewertung n>
"...."
```

Durch eine Methode können mehrere Symptome bewertet werden. In einer Wissensbasis kann eine Methode mehreren unterschiedlichen Symptomen zugeordnet sein. Dies ist erforderlich, da das Ergebnis einer Messung häufig mehrere Aussagen zugleich zuläßt. Die Symptombewertung trägt zunächst die Bewertung des Symptoms in die Agenda ein, dem die durchgeführte Methode zugeordnet war. Weitere, der Methode bekannte und durch sie bewertete Symptome werden mit Bewertung in die Agenda eingetragen, wenn die folgenden beiden Bedingungen zutreffen:

- Das Symptom gehört zum aktuellen Wissensbasismodul
- Das Symptom wurde korrekt bewertet ($-100 \leq BW(\text{Symptom}) \leq 100$)

Symptome, die bereits bewertet wurden, werden nicht neu bewertet, wenn keine zu große Abweichung zwischen aktuell ermittelter Bewertung und vorhandener Bewertung vorliegt. Gegebenenfalls wird der Anwender befragt.

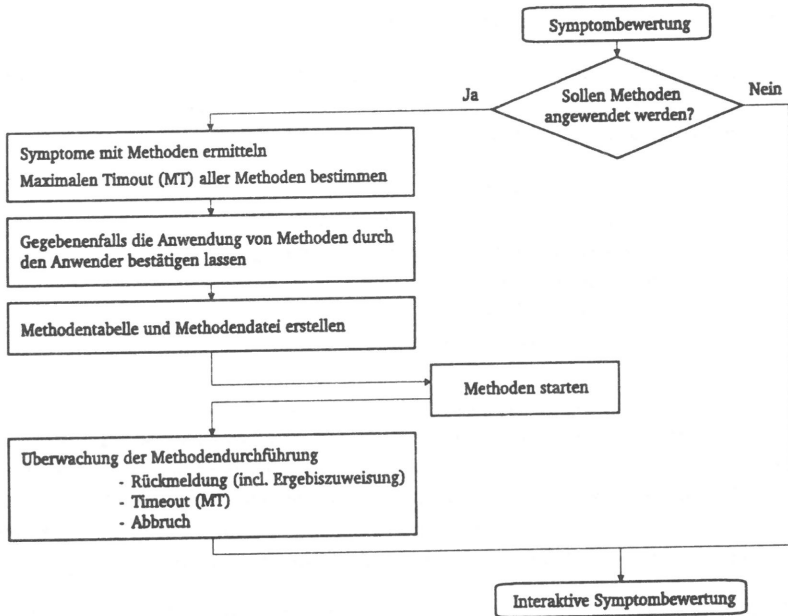


Bild 6.21 Flußdiagrammausschnitt der Agendaoperation "Symptombewertung" für die Durchführung von Methoden.

Durch Methoden können völlig neue Erkenntnisse über Fehlerursachen in die Agenda eingetragen werden. Wird im Verlauf der Problemlösung die Agendaoperation "Vorwärtsverkettung" durchgeführt, so kann dies zu einer Umorientierung der Problemlösung führen.

Nachdem alle Methoden gestartet wurden, wartet die Problemlösungskomponente auf eines der folgenden drei Ereignisse:

- Der maximale Timeout *MT* ist abgelaufen
- Alle Ergebnisdateien stehen zur Verfügung
- Der Anwender bricht die Bearbeitung ab

Ist der maximale Timeout abgelaufen, so werden die noch vorhandenen Prozesse zur Durchführung von Methoden abgeschlossen. Evtl. vorhandene Ergebnisse werden in die Agenda übernommen. Stehen alle erwarteten Ergebnisdateien zur Verfügung, dann wird die Bewertung von Symptomen durch Methoden abgeschlossen und die Ergebnisse werden in die Agenda übernommen. Durch die anschließende interaktive Bewertung von Symptomen werden evtl. noch unbewertete Symptome bewertet.

6.5.6 Bewertung der Problemlösungskomponente

Die Problemlösungskomponente des Diagnosesystems ist ein komplexes Softwaresystem, dessen Verhalten von vielen Parametern abhängt. Eine exakte, geschlossene Bewertung ist daher nicht möglich. Die Problemlösung wird wesentlich beeinflusst durch:

- Den Aufbau der Wissensbasis
- Die Einstellung der Problemlösungsparameter:
 - Die kompensatorischen Parameter λ_{und} und λ_{oder}
 - Die Diagnoseschwelle $TH_{Diagnose}$
 - Die Regelschwelle TH_{Regel}
- den Anwender, bei überwiegend interaktiver Symptombewertung

Im folgenden wird versucht, die Leistungsfähigkeit der Problemlösungskomponente anhand einer komplexeren Wissensbasis zu untersuchen. Die Wissensbasis besteht aus insgesamt 24 Wissensbasismodulen und enthält Diagnosewissen für das lokale Rechnernetz des Institutes für Nachrichtenvermittlung und Datenverarbeitung. Eine Komponente der vorgegebenen Wissensbasis wurde als defekte Komponente vorgegeben. Dementsprechend muß eine der betreffenden Komponente zugeordnete Enddiagnose eine maximale Bewertung erreichen.

Für die Bewertung der Symptome wurde anstelle eines Benutzers ein Zufallszahlengenerator verwendet, der gleichverteilt Zahlenwerte zwischen 0 und 1 liefert. Der Betrag des Bewertungswertes von Symptomen wurde mit einer stückweise linearen Funktion $R(x)$ bestimmt. Die Funktion $R(x)$ (Gl 6.14) ist definiert für das Intervall $x \in [0,1]$ und liefert für reelle x auch reelle Funktionswerte. Für die Bewertung werden die Funktionswerte von $R(x)$ gerundet, so daß natürliche Zahlen im Intervall $|BW(x)| \in [0,100]$ entstehen.

$$R(x) = \begin{cases} R(x) = 250 \cdot x & \text{für } 0 \leq x < 0,1 \\ R(x) = 18,75 + 62,5 \cdot x & \text{für } 0,1 \leq x \leq 0,9 \\ R(x) = -150 + 250 \cdot x & \text{für } 0,9 < x \leq 1 \end{cases} \quad (\text{Gl. 6.14})$$

Durch die Funktion $R(x)$ ist gewährleistet, daß 80% der Symptome mit Bewertungswerten aus dem Bewertungswertintervall $[25,75]$ bewertet werden. Jeweils 10% der Symptome werden aus den Intervallen $[0,25]$ und $(75,100]$ bewertet. Bild 6.22 stellt den Verlauf von $R(x)$ dar.

Für die positiv zur Problemlösung beitragenden Symptome wird der durch Zufall bestimmte Wert positiv als Bewertung übernommen, für die negativ zur Problemlösung beitragenden Symptome negativ. Symptome, die intern durch eine Regel bewertet werden können, werden durch die Regel bewertet. Der durch Zufall bestimmte Bewertungswert wird dann verworfen.

Tabelle 6.11 enthält die Ergebnisse von 25 Läufen für $\lambda_{und} = \lambda_{oder} = 0,6$ (linker Teil der Tabelle, Einstellung 1) und für $\lambda_{und} = \lambda_{oder} = 1$ (rechter Teil der Tabelle, Einstellung 2). Tabelle 6.12 faßt die wesentlichen Ergebnisse zusammen.

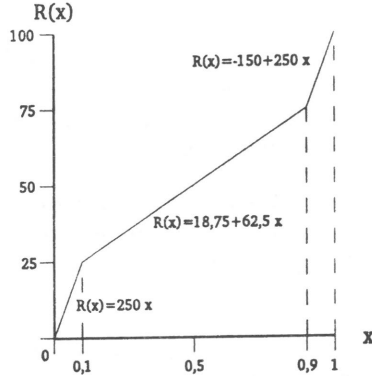


Bild 6.22 Schaubild der Funktion $R(x)$.

Ver- suchs Nr.	Einstellung 1: $\lambda_{und}=\lambda_{oder}=0,6$					Einstellung 2: $\lambda_{und}=\lambda_{oder}=1$				
	ge- funden	Be- wert- ung	An- zahl WBM	Bew. durch Datei	Bew. durch Regel	ge- funden	Be- wert- ung	An- zahl WBM	Bew. durch Datei	Bew. durch Regel
1	ja	57	3	9	0	ja	62	3	9	0
2	ja	46	4	10	0	ja	53	3	9	0
3	ja	59	3	9	0	ja	61	3	39	23
4	ja*	73	1	7	0	ja*	76	1	7	0
5	ja	40	11	70	25	ja	46	2	29	0
6	ja	47	4	10	0	ja	47	3	9	0
7	ja	63	3	9	0	ja	67	2	8	0
8	ja	55	12	80	25	ja	67	4	32	0
9	nein	(39)	3	9	0	ja	36	3	9	0
10	ja*	76	1	7	0	ja*	84	1	7	0
11	ja	60	4	10	0	ja*	71	1	7	0
12	ja	42	3	9	0	ja	47	2	8	0
13	ja	64	12	124	31	ja*	73	1	7	0
14	ja	42	4	10	0	ja	40	3	9	0
15	ja	59	9	54	23	ja	69	6	42	23
16	ja	36	4	10	0	ja	34	3	9	0
17	ja	68	2	8	0	ja	78	1	7	0
18	ja	53	19	202	41	ja	57	5	24	0
19	ja	48	4	10	0	ja	54	3	9	0
20	ja	40	4	10	0	ja	41	6	25	0
21	ja	57	6	51	23	ja	61	3	39	23
22	ja*	88	1	7	0	ja*	94	1	7	0
23	ja*	78	1	7	0	ja	95	1	7	0
24	nein		4	10	0	nein		3	9	0
25	ja	48	4	10	0	ja	50	2	8	0

Tabelle 6.11 Auflistung der Ergebnisse für Einstellung 1 ($\lambda_{und} = \lambda_{oder} = 0,6$) und Einstellung 2 ($\lambda_{und} = \lambda_{oder} = 1$).
* Frage, ob die angebotene Diagnose zutrifft, wurde mit ja beantwortet.
WBM Wissensbasismodul.

Bei Einstellung 2 ($\lambda_{\text{und}} = \lambda_{\text{oder}} = 1$) werden die kompensatorischen Operatoren für die Diagnosebewertung zum Minimum- bzw. Maximum-Operator (Siehe Abschnitt 6.2.3.2.5 und Abschnitt 6.2.3.2.6). Bei Einstellung 1 wird für die Bewertung der Diagnosen auch der Mittelwert, der zur Verknüpfung gehörenden Bewertungswerte berücksichtigt.

Bei Einstellung 2 wird in 24 von 25 Fällen die richtige Lösung gefunden. Bei Einstellung 1 nur in 23 Fällen. Zudem werden bei Einstellung 2 weniger Wissensbasen durchlaufen. Aus den exemplarisch dargestellten Tests geht hervor, daß bei exakt bewerteten Symptomen auch eine relativ exakte Diagnose gefunden wird.

In allen Fällen, in denen die vorgegebene Komponente bzw. die der Komponente zugeordnete Enddiagnose nicht als Lösung gefunden wurde, wurde immer eine der Komponente übergeordnete Komponente als Lösung angeboten. Das bedeutet, daß die Problemlösungskomponente zwar den fehlerhaften Bereich ausfindig machen konnte, nicht aber mit dem maximalen Detaillierungsgrad.

Die für die Bewertung der Problemlösungskomponente entwickelten Werkzeuge eignen sich sehr gut für den Test von Wissensbasismodulen. Anhand dieser Hilfsmittel kann geprüft werden, ob alle Enddiagnosen und Diagnosen eines Wissensbasismoduls durch die Problemlösungskomponente gefunden werden können und wie weit die einzelnen Lösungen in ihren Bewertungen auseinanderliegen.

	Einstellung 1 $\lambda_{\text{und}} = 0,6;$ $\lambda_{\text{oder}} = 0,6$	Einstellung 2 $\lambda_{\text{und}} = 1;$ $\lambda_{\text{oder}} = 1$
Anzahl Läufe mit gefundener Lösung	23	24
Anzahl der Läufe mit nicht gefundener Lösung	2	1
Durchschnittliche Anzahl von durchlaufenen Wissensbasismodulen	5,04	2,64
Durchschnittliche Anzahl von, mit dem Zufallszahlengenerator bewerteten, Symptomen	30,08	15
Durchschnittliche Anzahl anhand von Regeln bewerteter Symptome	6,72	2,76

Tabelle 6.12 Zusammenfassung der Ergebnisse der in Tabelle 6.11 dargestellten Tests.

6.6 Die Erklärungskomponente

6.6.1 Die Architektur der Erklärungskomponente

Die Erklärungskomponente hat die Aufgabe das Vorgehen und die Schlußfolgerungen der Problemlösungskomponente bei der Problemlösung zu erläutern. Ziel ist es, die Akzeptanz des Diagnosesystems durch den Anwender zu erhöhen [136]. Bild 6.23 zeigt ein Blockdiagramm der Architektur der Erklärungskomponente.

Um die Vorgehensweise des Diagnosesystems erklären zu können, müssen von der Erklärungskomponente alle Operationen der Problemlösungskomponente protokolliert werden (strategisches Wissen). Durch die Auflistung der durchgeführten Problemlösungsschritte wird die geforderte Transparenz der Problemlösung möglich. Die Kenntnis des Diagnoseverlaufes ist Grundvoraussetzung für die Erzeugung von Erklärungen. Die Plausibilität von Erklärungen wird erhöht durch die Angabe von Regeln und Prämissen, deren Bewertungen und den daraus gezogenen Schlußfolgerungen. Das hierfür notwendige Wissen kann den Wissensbasismodulen entnommen werden (strukturelles Wissen).

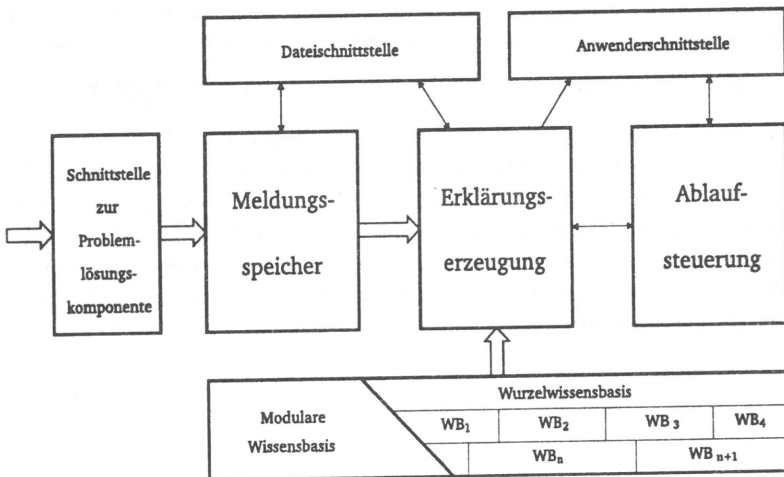


Bild 6.23 Architektur der Erklärungskomponente

Die Erklärungskomponente verfügt über 4 Schnittstellen. Das strategische Wissen wird über die Schnittstelle zur Problemlösungskomponente empfangen. Diese Schnittstelle wurde als Meldungsschnittstelle (Message-Queue) realisiert. Die Meldungen der Problemlösungskomponente werden im Meldungsspeicher abgelegt. Die Erklärungserzeugung verfügt über eine Schnittstelle zur modularen Wissensbasis. Über diese Schnittstelle kann das empfangene strategische Wissen

um strukturelles Wissen ergänzt werden. Die Anwenderschnittstelle dient der Interaktion mit dem Anwender. Über die Dateischnittstelle können erzeugte Erklärungen und Problem-lösungsprotokolle in Form von Dateien gespeichert werden.

Die über die Schnittstelle zur Problemlösungskomponente empfangenen Meldungen werden verkettet im Meldungsspeicher abgelegt. Die Erklärungserzeugung analysiert das Vorgehen der Problemlösungskomponente und erzeugt die gewünschten Erklärungen.

Die vom Meldungsspeicher empfangenen Meldungen können in ihrer ursprünglichen Form ebenfalls in einer Datei gespeichert werden. Damit wird die nachträgliche Dokumentation von Problemlösungen ermöglicht.

Die Erklärungskomponente erzeugt während einer Problemlösung:

- Erklärungen zum Stand der Problemlösung
- Erklärungen zu Fragen, die an den Anwender gestellt werden
- Erklärungen zu Wissensobjekten und deren Abhängigkeiten untereinander

Im Anschluß an eine Problemlösung kann ein Protokoll generiert werden, das alle Messungen, Symptome, Schlußfolgerungen usw. in geeigneter Form zusammenfaßt und dokumentiert.

6.6.2 Verbindung zwischen Problemlösungskomponente und Erklärungskomponente

Die Übertragung des strategischen Wissens von der Problemlösungskomponente an die Erklärungskomponente basiert auf Meldungen, die über eine Meldungswarteschlange (Message Queue) an die Erklärungskomponente weitergereicht werden. Jede Agendaoperation (siehe Abschnitt 6.5.2) der Problemlösungskomponente sendet eine entsprechende Meldung an die Problemlösungskomponente ab. Anhand der Meldungssequenz und ergänzendem Wissen aus den Wissensbasismodulen ist die Erklärungskomponente in der Lage, die Vorgehensweise der Problemlösungskomponente zu erläutern.

7. Zusammenfassung und Ausblick

7.1 Zusammenfassung

Im Rahmen der vorliegenden Arbeit wurde zunächst ein Einblick in die Problematik des Netzmanagement für heterogene lokale Netze gegeben. Der Stand der Normungsbestrebungen für das Management heterogener Rechnernetze wurde kurz dargestellt. Dabei wurde klar, daß die bisherigen Normungsbestrebungen nur die Datenerfassung und Datenmanipulation von Netzressourcen erfassen. Die Entwicklung von Systemen, die den Anwender bei der Aufgabe der Steuerung, Überwachung und Diagnose von Netzen unterstützen, bleibt offen.

Für das Netzmanagement, oder für Teilbereiche davon, wurde die Anwendung wissensbasierter Systeme diskutiert. Auf existierende Prototypen aus dem Bereich der Telekommunikationsnetze wurde hingewiesen.

Für lokale Rechnernetze, speziell für den Bereich der Fertigungsautomatisierung, wurde ein System für die Fehlerdiagnose entwickelt. Nach einer Einführung in die Systemarchitektur wurden die Teilsysteme:

- Verteiltes Meßsystem für lokale Rechnernetze nach IEEE 802.3 und
- Diagnosesystem

detailliert beschrieben. Nochmals erwähnt werden soll, daß das vorgestellte verteilte Meßsystem unter anderem die Durchführung von Reflektionsmessungen bei Netzbetrieb erlaubt und daß sich das Diagnosesystem, aufgrund der Modularisierung der Wissensbasis, sehr gut für die Diagnose großer, komplexer Systeme eignet. Es wurde Wert auf eine offene, durch den Anwender erweiterbare, Systemarchitektur gelegt. Ferner wurde ein Ansatz für Test und Beurteilung der Problemlösungskomponente erläutert.

Das vorgestellte System entstand in einem Projekt, das im Auftrag des Bereiches Automatisierungstechnik der Firma Siemens am Institut für Nachrichtenvermittlung und Datenverarbeitung durchgeführt wurde. Prototypen des Systems befinden sich im Hause Siemens und am Institut für Nachrichtenvermittlung und Datenverarbeitung im Einsatz. Diagnosesystem und Monitoringsystem wurden in der Programmiersprache C unter UNIX entwickelt [59, 107]. Ursprünglich war die direkte Kopplung von verteiltem Meßsystem und Diagnosesystem geplant. Ein erster Prototyp mit dieser zweigeteilten Systemarchitektur wurde 1990 Anwendern übergeben. Dem Wunsch der Anwender, selbst das verteilte Meßsystem mit seinen Meßmöglichkeiten bedienen zu können und evtl. andere fernsteuerbare Meßgeräte in das System einbinden zu können, wurde durch das Hinzufügen des Monitoringsystems entsprochen.

Ein Anwendungsfall aus der Praxis sei hier noch erwähnt. Sich verändernde physikalische Eigenschaften des Übertragungsmediums bewirken unter Umständen eine Verschlechterung der Eigenschaften eines lokalen Netzes. So sind z.B. Quetschungen an Segmentkabeln eine Ursache von minimal erhöhten Reflektionen. Diese wirken sich besonders bei der Übertragung größerer Datenpakete aus. Die Funktionalität des lokalen Netzes wird dadurch nicht unbedingt so beein-

trächtig, daß dies der Anwender bemerkt. Fehlerhaft übertragene Datenpakete werden durch Protokolle höherer Schichten erkannt und wiederholt. Lediglich die Leistung des Netzes sinkt, was der Anwender sich häufig mit einer gestiegenen Netzlast erklärt. Derartige Veränderungen an Netzkabeln (vor allem bei Anwendung von 10Base2 (Cheapernet)) wurden recht häufig durch das System durch die Auswertung automatisch bei Netzbetrieb durchgeführter Reflektionsmessungen erkannt.

7.2 Ausblick

Die Arbeiten zum Monitoringsystem sind noch nicht abgeschlossen und werden weitergeführt. Ferner wird die Implementierung einer OSI konformen Protokollarchitektur für das verteilte Meßsystem erwogen. Die möglichen Messungen sollen entsprechend in Managed Objects abgebildet werden.

Ferner sollen Netzwerk Management Produkte anderer Hersteller an das Diagnosesystem zur Erweiterung der Möglichkeiten, automatisch fehlerabhängiges Wissen zu erfassen, angeschlossen werden.

Im Bereich der Wissenserfassung ist geplant, die Grundstruktur einer Netzbeschreibung automatisch aus der Information des Directory Service [86-93] zu erzeugen. Kann damit die Wurzelwissensbasis für ein lokales Rechnernetz generiert werden und stehen die speziellen Wissensbasismodule für verwendete Systeme zur Verfügung, so verringert sich der Aufwand für die Erstellung einer Wissensbasis für ein lokales Rechnernetz vermutlich beträchtlich.

Das Diagnosesystem wird bislang mit einer am Institut für Nachrichtenvermittlung und Datenverarbeitung entwickelten semigraphischen Oberfläche [122] betrieben. An einer Umstellung auf die OSF/Motif Oberfläche wird gearbeitet [12, 108, 170].

Für verteilte Performance Messungen von Protokollimplementierungen wird das verteilte Meßsystem in einem anderen Projekt in modifizierter Form verwendet [187].

Literaturverzeichnis

- [1] Aidarous, S., E.; Helmy, D.: Integration of Expert Systems into Network Operations and Management. *Proceedings of the IEEE ICC'89*, Seiten 1438-1442, 1989.
- [2] Ali, S.: *Implementierung und Test von Problemlösungsalgorithmen für ein Diagnose-Expertensystem für lokale Netze*. Studienarbeit Nr. 1098, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1991.
- [3] Ashany, R.; Ferrari, D.; Pasquale, J.: Application of AI-Techniques to Adaptive Routing in Wide Area Networks. *Proceedings of the Seventh Annual International Phoenix Conference on Computers and Communications*, Scottsdale, Seiten 157-160, March 1988.
- [4] Astrath, K.-H.: Das Übertragungsmedium Kabel. *DATACOM*, 6. Jahrgang, Heft 2, Seiten 130-136, Februar 1989.
- [5] Balzer, F.: *Entwurf und Realisierung einer Auswertesoftware für ein Reflektionsmeßgerät*. Diplomarbeit Nr. 1127, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1992.
- [6] Barker, V.; O'Connor, D., E.: Expert Systems for Configuration at Digital: XCON and Beyond. *Communications of the ACM*, Vol. 32, Number 6, Seiten 298-318, March 1989.
- [7] Barth, G.: Prolog: Programmierung auf der Basis von Logik. *it - Informationstechnik*, 29. Jahrgang, Heft 4, April 1987.
- [8] Beck, J.: *Automatische Reflektionsmessung in LAN Segmenten nach IEEE 802.3*. Diplomarbeit Nr. 953, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1989.
- [9] Ben-Artzi, A.; Chandna, A.; Warriar, U.: Network Management of TCP/IP Networks: Present and Future. *IEEE Network*, Vol. 4, No. 4, Seiten 35-42, July 1990.
- [10] Bender, K.: Profibus. Carl Hanser Verlag, München, Wien, 1990.
- [11] Berberich, M.; Feicht, E.; Kwee-Christoph, E.; Lauer, T.; Lehmann, A.; Stürner, T.: An Expert System Approach for Maintenance Support of EWSD Switches. *Proceedings of the IEEE ICC'89*, Seiten 1433-1439, 1989.
- [12] Berlage, T.: OSF, Motif und das X-Window-System. Addison-Wesley, München, Reading, 1991.
- [13] Bernstein, L.; Yuhas, C.,M.: Expert Systems in Network Management - The Second Revolution. *IEEE Selected Areas in Communications*, Vol. 5, No. 5, Seiten 784-787, Juni 1988.
- [14] Bönsch, W.: *Aufbau und Integration eines automatischen Reflektionsmeßgerätes für LAN Segmente nach IEEE 802.3*. Diplomarbeit Nr. 1051, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1990.
- [15] Bosch, M.: *Kopplung von Kommunikationsnetzen: Architekturen, Leistungsuntersuchungen und eine Beispielrealisierung*. 52. Bericht über verkehrstheoretische Arbeiten (Dissertationschrift), Universität Stuttgart, Institut für Nachrichtenvermittlung und Datenverarbeitung, 1992.
- [16] Boumiza, M.: *Online-Hilfesystem für ein Diagnosesystem*. Studienarbeit Nr. 1096, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1991.
- [17] Bratko, I.: Prolog Programming for Artificial Intelligence. Addison-Wesley, Reading, 1986.
- [18] Brown, K., E.; Cowan, C., F., N.; Crawford, T., M.; Grant, P., M.: Knowledge Based Techniques for Fault Detection in Digital Microwave Radio Communication Equipment. *IEEE Selected Areas in Communications*, Vol. 6, No. 5, Seiten 819-827, Juni 1988.

- [19] Brenner, J.: *Implementierung und Test von Problemlösungsalgorithmen für heterarchische Wissensbasen*. Diplomarbeit Nr. 1095, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1991.
- [20] Brusil, P., J.; Stokesberry, D., P.: Toward a unified theory of managing large networks. *IEEE Spectrum*, Vol. 26, No. 4, Seiten 39-42, April 1989
- [21] Buchanan, B.; Shortliffe, E.: *Rule Based Expert Systems - The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison Wesley, Reading, 1984.
- [22] Bullinger, H.-J.; Fähnrich, K.-P. und andere: *Expertensysteme*. Expert Verlag, Ehningen bei Böblingen, 1988.
- [23] Callahan, P., H.: Expert Systems for AT&T Switched Network Maintenance, *AT&T Technical Journal*, Vol. 67, No. 1, Seiten 93-103, Januar/Februar 1988.
- [24] Cassel, L., N.; Amer, P., D.: Management of Distributed Measurement Over Interconnected Networks. *IEEE Network*, Vol. 2, No. 2, Seiten 50-55, 1988.
- [25] Chylla, P.; Hegering, H.-G.: *Ethernet-LANs*. Datacom Buchverlag, Pulheim, 1987.
- [26] Cohen, R., S.; Kan, H., K.; Pennotti, R. J.: Unified Network Management from AT&T. *AT&T Technical Journal*, Vol. 67, No. 6, Seiten 121-137, November/December 1988
- [27] Cochran, T.; Mellichamp, J., M.: AUTOREC: An Automated Error Recovery System for Network Management. *IEEE Network*, Vol. 4, No.3, Seiten 14-18, März 1990.
- [28] Cronk, R., N.; Callahan, P., H.; Bernstein, L.: Rule-Based Expert Systems for Network Management Operations: An Introduction. *IEEE Network*, Vol. 2, No. 5, Seiten 7-21, September 1988.
- [29] Dal Cin, M.; Philip, T.: Expertensysteme für die Fehlerdiagnose. *it - Informationstechnik*, 30. Jahrgang, Heft 4, Seiten 237-246, April 1988.
- [30] Decker, R.: Grundlegende Literatur zu Themen der Wissensrepräsentation. *Informationstechnik - it*, Oldenbourg Verlag, 31. Jahrgang, Heft 2, Seiten 167-170, Februar 1989.
- [31] Delfs, H.: Diagnose-Expertensysteme brauchen Hypertext - Das Beispiel MAX. *Proceedings zum Workshop "Hypertext/Hypermedia 90" der Gesellschaft für Mathematik und Datenverarbeitung (GMD)*, Springer Verlag, 1990.
- [32] Dermla, A.; Marcus, O.-W.; Strassacker, P.: Ein globales Projektier- und Managementwerkzeug für lokale Netze, *Automatisierungstechnische Praxis atp*, 32. Jahrgang, Heft 5, Seiten 258-261, Mai 1990.
- [33] Dörr, T.: *Integration eines LAN-Durchflußmessers in ein LAN-Meßsystem für lokale Netze nach IEEE 802.3*. Studienarbeit Nr. 1100, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1991.
- [34] Düpmann, F.: *Aufbau einer Echobox für Ethernet LANs*. Diplomarbeit Nr. 1048, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1990.
- [35] Eberspächer, M.: *Mustererkennung bei digitalen Reflektionsmessungen mit neuronalen Netzen*. Studienarbeit Nr. 1049, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1990.
- [36] Eckardt, K.-J.; Nowak, R.: *Standard-Architekturen für Rechnerkommunikation*, Handbuch der Informatik, Band 7.1. R. Oldenbourg Verlag, München, Wien, 1988.
- [37] Elsam, E.; Mayersohn, J.: Using AI to Plan the Defense Data Network. *Defense Electronics*, Seiten 175-184, June 1987.

- [38] Embry, J.; Manson, P.; Milham, D.: An Open Network Management Architecture: OSI/NM Forum Architecture and Concepts. *IEEE Network*, Vol. 4, No. 4, Seiten 14-22, Juli 1990.
- [39] Endres, R.: *Entwurf und Integration eines Meßprotokolls für ein verteiltes LAN-Meßsystem*. Diplomarbeit Nr. 1090, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1991.
- [40] Endres, R.; Kempf, R.; Schmid, M.; Schöllkopf, T.; Schröder, J.: Meßmöglichkeiten des modularen, verteilten LAN-Meßsystems. Internes Arbeitspapier, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart., Januar 1991.
- [41] Estenfeld, K.: Prolog-Implementierung, Konzepte und Realisierungen. *Informatik-Spektrum*, Springer-Verlag, Band 10, Heft 10, Seiten 67-78, Oktober 1987.
- [42] ETSI Draft: ETR on objectives, principles, concepts and reference configurations for a Telecommunications Management Network. Doc ETSI/TC NA (91)), Strasbourg, Mai 1991.
- [43] Fahlman, S., E.; Lebiere, C.: The Cascade-Correlation Learning Architecture. School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, February, 14, 1990.
- [44] Feridun, M.; Leib, M.; Nodine, M.; Ong, J.: ANM: Automated Network Management System. *IEEE Network*, Vol.2, No. 2, Seiten 13-19, 1988.
- [45] Fink, P., K.; Lusth, J., C.; Duran, J., W.: A General Expert System Design for Diagnostic Problem Solving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 7, No. 5, Seiten 553-560, September 1985.
- [46] Fox, J., R.; Slawsky, G., M.: Concept Prototyping a Real Time Switch Maintenance Expert System. *IEEE ICC'88 Conference Record*, 1988.
- [47] Fox, J., R.; Slawsky, G., M.: The Role of Expert Systems in Switch Maintenance Operations and the Generation of Switch Analysis Requirements. *IEEE Journal on Selected Areas in Communications*, Vol. 6, No. 4, Seiten 706-714, März 1988
- [48] Fujimoto, K.; Hibino, Y.; Oimatsu, T.: Network Operations Expert System. *Review of the Electrical Communications Laboratories*, Vol. 37, No. 1, Seiten 15-21, Januar 1989.
- [49] Gemkow, U., Schröder, J., Weixler, M.: Betriebssoftware des LAN-Meßrechners. Internes Diskussionspapier, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, Januar 1990.
- [50] Gerblinger, D.: *Integration eines Kommunikationssystems für ein verteiltes LAN-Meßsystem*. Diplomarbeit Nr. 1149, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1992.
- [51] Gühr, O.; Weixler, M.: Messung der Datenverkehrsprofile in lokalen Netzen. *Proceedings Kommunikation in verteilten Systemen*, Seiten 861-877, Stuttgart, 22.-24. Februar 1989.
- [52] Giovannini, F.; Rossi, C., F.: An Expert System for the Management of the Italian Public Packet-Switched Network. *Proceedings of the IEEE ICC'89*, Seiten 1428-1432, 1989
- [53] Götzer, M.: *Systemintegration eines modularen LAN-Meßsystems für lokale Netze nach IEEE 802.3*. Diplomarbeit Nr. 1094, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1991.
- [54] Graf, M.: *Mustererkennung bei digitalen Reflektionsmessungen*. Studienarbeit Nr. 972, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1989.

- [55] Gupta, A.; Banderredì, E., P. (Editors): Principles of Expert Systems. A volume in the IEEE PRESS Selected Reprint Series, IEEE, 1988.
- [56] Gupta, A.; Banderredì, E., P. (Editors): Microcomputer-Based Expert Systems. A volume in the IEEE PRESS Selected Reprint Series, IEEE, 1988.
- [57] Habel, Ch.: Repräsentation von Wissen. *Informatik Spektrum*, Springer-Verlag, 13. Jahrgang, Seiten 126-136, 1990.
- [58] Härder, T.: Klassische Datenmodelle und Wissensrepräsentation. *Informationstechnik - it*, Oldenbourg Verlag, 31. Jahrgang, Heft 2, Seiten 141-154, Februar 1989.
- [59] Harbison, S., P.; Steele, Guy, L.: C A Reference Manual. Printice-Hall, Englewood Cliffs, New Jersey, USA, 1987.
- [60] Halsall, F.; Modiri, N.: Protocol analyser for the monitoring and analysis of OSI networks. *Computer Communications*, Vol. 13, No. 9, Seiten 533-541, November 1990.
- [61] Halsall, F.; Modiri, N.: An Implementation of an OSI Network Management System. *IEEE Networks*, Vol. 4, No. 4, Seiten 44-53, Juli 1990.
- [62] Harmon, P.; King, D.: Expertensysteme in der Praxis. Oldenbourg Verlag, 3. Auflage, München, Wien 1989
- [63] Hegering, H.-G.; Valta, R.: Netzmanagement - Aufgaben und Architekturkonzepte. *Tutorium der 6. GIITG-Fachtagung "Kommunikation in verteilten Systemen"*, Stuttgart, 20.-24. Februar 1989.
- [64] Heigert, J.: OSI-Netzwerk-Management: Status und Perspektiven. *DATACOM*, 7. Jahrgang, Heft 1, Januar 1990.
- [65] Held, H.-J.: Verfügbarkeitssteigerung von Produktionseinrichtungen durch wissensbasierte Fehleranalyse. *Automatisierungstechnische Praxis - atp*, 32. Jahrgang, Heft 5, Seiten 248-257, Mai 1990.
- [66] Henne, P.; Klar, W.; Wittur, K.-H.: Ein Expertensystem zur Fehlerdiagnose im automatischen Getriebe. *Proceedings GI-Kongress: Wissensbasierte Systeme*, Informatik Fachberichte Nr. 112, Springer-Verlag, Heidelberg, Seiten 105-120, 1985.
- [67] Hiebert, L.: AI and Network Planning, *AI Expert*, Vol. 3, No. 9, Seiten 26-33, September 1988.
- [68] Hinz, T.: *Systemaufbau und Integration eines Meßsystems für lokale Netze nach IEEE 802.3*. Diplomarbeit Nr. 1052, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1990.
- [69] IEEE 802.2: Local Area Networks - Logical Link Control, 1985.
- [70] IEEE 802.3: Local Area Networks - CSMA/CD Access Method and Physical Layer Specification, 1985.
- [71] IEEE 802.4: Local Area Networks - Token Passing Bus Access Method and Physical Layer Specification, 1985.
- [72] IEEE 802.5: Token Passig Bus Access Method and Physical Layer Specification, 1985
- [73] Intel: LAN Components User Manual, Intel , 1984.
- [74] Intel: 16-Bit Embedded Controller Handbook, Intel, 1989.
- [75] Isermann, R.: Identifikation dynamischer Systeme. Band I+II. Springer-Verlag, Berlin, Heidelberg, New York, 1988.

- [76] ISO 7498: Information Processing Systems - Open System Interconnection, Basic Reference Model. International Standard, 1984.
- [77] ISO 7498/4: Information Processing Systems - Open System Interconnection, Basic Reference Model - Part 4: OSI Management Framework. International Standard, 1989.
- [78] ISO 8571: File Transfer, Access and Management. International Standard, 1988.
- [79] ISO 8649: Service Definition for the Association Control Service Element. Draft International Standard, 1988.
- [80] ISO 8650: Protocol Specification for the Association Control Service Element. Draft International Standard, 1988.
- [81] ISO 8802/2: Local Area Networks - Logical Link Control. International Standard, 1988.
- [82] ISO 8802/3: Local Area Networks - CSMA/CD Access Method and Physical Layer Specification. International Standard, 1989.
- [83] ISO 8802/4: Local Area Networks - Token Passing Bus Access Method and Physical Layer Specification. Draft International Standard, 1988.
- [84] ISO 8802/5: Token Passig Bus Access Method and Physical Layer Spezifikation. Draft Proposal, 1985.
- [85] ISO 9506: Manufacturing Message Specification. International Standard, 1989.
- [86] ISO 9594/1: The Directory-Part 1: Overview of Concepts, Models and Service. Draft International Standard, 1987.
- [87] ISO 9594/2: The Directory-Part 2: Models. Draft International Standard, 1987.
- [88] ISO 9594/3: The Directory-Part 3: Abstract Service Definition. Draft International Standard, 1987.
- [89] ISO 9594/4: The Directory-Part 4: Procedures for Distributed Operations. Draft International Standard, 1987.
- [90] ISO 9594/5: The Directory-Part 5: Protocol Specifications. Draft International Standard, 1987.
- [91] ISO 9594/6: The Directory-Part 6: Selected Attribute Types. Draft International Standard, 1987.
- [92] ISO 9594/7: The Directory-Part 7: Selected Object Classes. Draft International Standard, 1987.
- [93] ISO 9594/8: The Directory-Part 8: Authentication Framework. Draft International Standard, 1987.
- [94] ISO 9595/1: Common Management Information Service. Draft International Standard, 1988
- [95] ISO 9596/2: Common Management Information Protocol. Draft International Standard, 1988.
- [96] Jackson, P.: Expertensysteme. Addison-Wesley Verlag, Bonn, 1987
- [97] Joseph, C., A.; Muralidhar, K., H.: Integrated Network Management in an Enterprise Environment, *IEEE Network*, Vol. 4, No. 4, Seiten 7-13, Juli 1990.
- [98] Kauffels, F.-J.: Netzwerk-Management, Teil 1: Einführende Bestandsaufnahme und Ausblick. *DATAKOM*, 6. Jahrgang, Heft 3, Seiten 98-104, März 1989.
- [99] Kauffels, F.-J.: Netzwerk-Management, Teil 2: Abschluß der Bestandsaufnahme. *DATAKOM*, 6. Jahrgang, Heft 4, Seiten 70-75, April 1989.

- [100] Kauffels, F.-J.: Netzwerk-Management, Teil 3: Management in OSI-Umgebungen. *DATAKOM*, 6. Jahrgang, Heft 5, Seiten 78-83, Mai 1989.
- [101] Kauffels, F.-J.: Netzwerk-Management, Teil 4: Netzmanagement in SNA, grundsätzliche Struktur. *DATAKOM*, 6. Jahrgang, Heft 6, Seiten 82-88, Juni 1989.
- [102] Kauffels, F.-J.: Netzwerk-Management, Teil 5: Netzmanagement in SNA: Netview. *DATAKOM*, 6. Jahrgang, Heft 8, Seiten 80-84, August 1989.
- [103] Kauffels, F.-J.: Netzwerk-Management, Teil 6: Netzmanagement bei DEC und Siemens. *DATAKOM*, 6. Jahrgang, Heft 9, Seiten 122-126, September 1989.
- [104] Kauffels, F.-J.: Netzwerk-Management, Teil 7: AT&Ts Unified Network Management Architecture UNMA, Management Foren, Integratoren und Direktoren. *DATAKOM*, 6. Jahrgang, Heft 11, Seiten 80-87, November 1989.
- [105] Kauffels, F.-J.: Netzwerk-Management, Teil 8: DEC-Enterprise Management Architecture EMA. *DATAKOM*, 7. Jahrgang, Heft 1, Seiten 82-86, Januar 1990.
- [106] Kempf, R.: *Entwurf und Implementierung einer Verkehrsflußmeßkomponente für ein verteiltes LAN-Meßsystem*. Studienarbeit Nr. 1118, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1991.
- [107] Kernighan, B., W.; Pike, R.: *The UNIX Programming Environment*. Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1984.
- [108] Keusch, R.: *Portierung eines Diagnose-Expertensystems auf UNIX/386 und OSF/Motif*. Diplomarbeit Nr. 1157, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1991.
- [109] Khan, N., A.; Callahan, P., H.; Dube, R.; Tsay, J., L.; Van Dusen, W.: *An Engineering Approach to Model-Based Troubleshooting in Communication Networks*. *IEEE Selected Areas in Communications*, Vol. 6, No. 5, Seiten 792-800, Juni 1988.
- [110] Kiesel, W.: Netzwerk-Diagnose in der Produktionsautomatisierung. *DATAKOM*, 6. Jahrgang, Heft 12, Seiten 38-41, Dezember 1989.
- [111] Kiesel, W.: Netzwerkmanagement für Kommunikationsnetze in der Produktionsautomatisierung. *Proceedings Kommunikation in verteilten Systemen*, Seiten 718-729, Stuttgart, 22.-24. Februar 1989.
- [112] Klerer S., M.: *The OSI Management Architecture: an Overview*. *IEEE Network*, Vol. 2, No. 2, Seiten 20-29, 1988.
- [113] Köpke, D.: Fehlererkennung auf dem Ethernet. *DATAKOM*, 6. Jahrgang, Heft 9, Seiten 22-28, September 1989.
- [114] Koseki, Y.; Wada, S.-i.; Nishida, T.; Mori, H.: SHOOTX: A Multiple Knowledge-based Diagnosis Expert System for NEAX61 ESS. *IEEE ISS Conference Record*, 1987.
- [115] Kühn, P., J.; Pritschow, G. (Bandherausgeber): *Kommunikationstechnik für den rechnerintegrierten Fabrikbetrieb*. Springer-Verlag, Verlag TÜV Rheinland, 1991.
- [116] Kühn, P., J.; Rößler, G.; Schröder, J.: *Kommunikationsnetze als Bindeglied verteilter Steuerungen in der Fabrik*. *Tagungsband des Fertigungstechnischen Kolloquiums (FTK) '91*, Springer-Verlag, S.48-51, 1991.
- [117] Kunigk, H.-W.; Stauß, G.: *Vernetzungsaspekte im Fertigungsberreich*. *DATAKOM*, 9. Jahrgang, Heft 1, Januar 1992.

- [118] Kurrek, K.: Die Integration der Fertigung auf der Basis von Ethernet LANs. *DATAKOM*, 6. Jahrgang, Heft 6, Seiten 98-104, Juni 1989.
- [119] Laicher, J.: *Aufbau eines Durchflußmessers für Ethernet-Transceiverkabel*. Studienarbeit Nr. 1050, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1990.
- [120] Lauber, R.: *Prozeßautomatisierung*, Band 1 & 2, Springer-Verlag, Berlin, 1989.
- [121] Laufer, W.: *Test und Leistungsbewertung eines Diagnosesystems für Lokale Netze*. Diplomarbeit Nr. 1130, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1992.
- [122] Lederer, A.: *Aufbau einer Expertensystemshell für ein modulares verteiltes Diagnosesystem auf dem Betriebssystem UNIXIXENIX*. Diplomarbeit Nr. 1046, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1990.
- [123] Leung, K., S.; Lam, W.: Fuzzy Concepts in Expert Systems. *IEEE Computer*, Vol. 21, No. 9, Seiten 43-56, September 1988.
- [124] Liu, D.-M., D.; Pelz, D., A.: I-Test: Integrated Testing Expert System for Trunks. *IEEE Selected Areas in Communications*, Vol. 6, No. 5, Seiten 800-804, Juni 1988.
- [125] Lohberg, G.: "SMART II: Knowledge Requirements for Expert Systems. *IEEE ICC'88 Conference Record*, 1988.
- [126] Lohmiller, R.: *Entwurf und Implementierung eines Monitoringsystems für ein verteiltes LAN-Meßsystem*. Diplomarbeit Nr. 1079, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1992.
- [127] Maetschke, S.: *Analyse von TDR-Messungen mit Methoden der Neuroinformatik*. Diplomarbeit, Fachbereich Informatik, Fachhochschule Würzburg-Schweinfurt, 1990.
- [128] Marques, T., E.: A Symptom-Driven Expert System for Isolating and Correcting Network Faults. *IEEE Communications Magazine*, Vol. 26, No. 3, Seiten 6-14, March 1988.
- [129] MAP: MAP Network Architecture. MAP 3.0 Implementation Release, General Motors Technical Center, 1987.
- [130] Mathonet, R; Van Cotthem, H.; Vanryckeghem, L.: DANTEs: An Expert System for Real Time Network Troubleshooting. *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Seiten 527-530, Milano, Italy, August, 1987.
- [131] McClelland, J., L.; Rumelhart D., E.: *Explorations in Parallel Distributed Processing*. MIT Press, Cambridge, Massachusetts, London, England, 1988.
- [132] McClelland, J., L.; Rumelhart, D.,E. and the PDP Research Group: *Parallel Distributed Processing, Volume 2: Psychological and Biological Models*. MIT Press, Cambridge, Massachusetts, London, England, 1988.
- [133] Mescheder, B.; Westernhoff, T.: *Offene Architekturen in Expertensystem-Shell*s. *Angewandte Informatik*, Vieweg Verlag, Heft 9, 1988.
- [134] MIL-STD-1777: Internet Protocol. Department of Defense, USA, November 1983.
- [135] MIL-STD-1778: Transmission Control Protocol. Department of Defense, USA, November 1983.
- [136] Moosburger, T.: *Entwurf und Implementierung einer Erklärungskomponente für ein Diagnosesystem*. Diplomarbeit Nr. 1093, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1991.

- [137] Muralidhar, K., H.; Irish, B., W.: MAPCON: An Expert System for Configuration of MAP Networks. *IEEE Selected Areas in Communications*, Vol. 6, No. 5, Seiten 869-874, Juni 1988.
- [138] Mutter, M.: *Betriebssystem für ein modulares Meßsystems*. Diplomarbeit Nr. 1085, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1991
- [139] Orleth, R.: *Floppy-Disk Controller für ein verteiltes Meßsystem*. Studienarbeit Nr. 1099, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1991.
- [140] Orleth, R.: *Integration der Softwarekomponenten eines verteilten LAN-Meßsystems*. Diplomarbeit Nr. 1148, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1992.
- [141] Pagurek, B.; Kaye, A., R.; Helmy, D.: Knowledge Based Fault Location in a Data Communication Network, IEEE document CH2538-7/88/0000-1729, Seiten 1729-1733.
- [142] Peacocke, D.; Rabie, S.: Knowledge Based Maintenance in Networks. *IEEE Selected Areas in Communications*, Vol. 6, No. 5, Seiten 813-818, Juni 1988.
- [143] Peacocke, D.; Giuffrida, M.: Big Brother: A Network Service Expert. *Proceedings of the 1988 Summer USENIX Conference*, USENIX Association, San Francisco, Seiten 393-398, June 20-24, 1988.
- [144] Permantier, G.: *Verfahren der Wissensrepräsentation bei der Entwicklung wissensbasierter Systeme für die Automatisierungstechnik*. Dissertationschrift, Universität Stuttgart, Institut für Regelungstechnik und Prozeßautomatisierung, 1991.
- [145] Pfaffhausen, H.: Realisierung eines regelbasierten Expertensystems durch einen Logiksimulator. *Informationstechnik - it*, Oldenbourg Verlag, 32. Jahrgang, Heft 4, Seiten 276-280, April 1990.
- [146] Pichler, S.; Graf, V.: Strukturierte Implementierung von IEEE 802.3 Netzen. *DATACOM*, 6. Jahrgang, Heft 10, Seiten 82-85, Oktober, 1989.
- [147] Poyhonen, P. et al.: Multimodel Reasoning in DX200 Digital Switching System Diagnosis. *IEEE ISS Conference Record*, 1987.
- [148] Prasser, K.: Unscharfes Schließen in wissensbasierten Systemen. *Informationstechnik - it*, Oldenbourg-Verlag, 34. Jahrgang, Heft 2, Seiten 124-131, April, 1992.
- [149] Puppe, F.: Diagnostisches Problemlösen mit Expertensystemen. Springer Verlag, Informatik-Fachberichte Nr. 148, Berlin. Heidelberg, New York, 1987.
- [150] Puppe, F.: Einführung in Expertensysteme. Springer Verlag, Studienreihe Informatik, Berlin. Heidelberg, New York, 1988.
- [151] Puppe, F.: Diagnostik-Expertensysteme. *Informatik Spektrum*, Springer-Verlag, Band 10, Heft. 10, Seiten 294-308, Oktober 1987.
- [152] Rabie, S.; Bult, T.; Carter, V., S.; Cavan, B.: DRESS: Diagnostic Repair Expert System Shell. *IEEE ICC'88 Conference Record*, 1988.
- [153] Rabie, S.; Rau-Chaplin, A.; Shibahara, T.: DAD: A Real-Time Expert System for Monitoring of Data Paket Networks. *IEEE Network*, Vol. 2, No. 5, Seiten 29-34, September, 1988.
- [154] Reinfrank, M.: Formeln und Modelle: Wissensrepräsentation mit Logik. *Informationstechnik - it*, 31. Jahrgang, Heft 2, Seiten 102-112, Februar, 1989.
- [155] Reinhold, W.: *Modulares verteiltes Diagnosesystem*. Diplomarbeit Nr. 940, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1988.

- [156] Rommelfanger, H.: Entscheiden bei Unschärfe. Springer-Verlag, Hochschultext, Berlin Heidelberg New York, 1988.
- [157] Roos, B.: *Implementierung von LAN-Hardware-Fehlerszenarien in der Sprache PROLOG*. Studienarbeit Nr. 864, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1988.
- [158] Roos, J.; Jordaán, D.; Markgraaff, M.; van Rooyen, F.: Model and architecture of a generalized network management system. *Computer Communications*, Vol. 13, No. 9, Seiten 517-526, November 1990.
- [159] Rose, O.: Ein Überblick über die ISO/OSI-Management-Architektur. *PIK Praxis der Informationsverarbeitung und Kommunikation*, 12. Jahrgang, Heft 3, Seiten 150-159, 1989
- [160] Ross, M., J.; Covo, A., A.; Hart, C., D., Jr.: An AI-Based Network Management System. *Proceedings of the Seventh Annual International Phoenix Conference on Computers and Communications*, Scottsdale, Seiten 458-461, March 1988.
- [161] Rumelhart, D.E.; McClelland, J., L. and the PDP Research Group: Parallel Distributed Processing, Volume 1: Foundations. MIT Press, Cambridge, Massachusetts, London, England, 1988.
- [162] Sahin, V.; Omidyar, C., G.; Bauman, T., M.: Telecommunications Management Network (TMN) Architecture and Interworking Designs. *IEEE Journal on Selected Areas in Communications*, Vol. 6, No. 4, May 1988.
- [163] Salomon, R.: Beschleunigtes Lernen durch adaptive Regelung der Lernrate bei Back-propagation in feed forward Netzen. *Proceedings der 6. Österreichischen Artificial-Intelligence-Tagung (KONNAI)*, Salzburg, Springer-Verlag, Berlin, Heidelberg, Seiten 173-178, 18.-21. September 1990.
- [164] Sameh, R.; Rau-Chaplin, A.; Shibahara, T.: DAD: A Real-Time Expert System for Monitoring of Data Packet Networks. *IEEE Network*, Vol. 2, No. 5, September, 1988.
- [165] Savory, S. E.: Grundlagen von Expertensystemen. Oldenbourg Verlag, München, 1990.
- [166] Sayles, W.; Thomas, J.: Finding and fixing network faults with an expert system. *Data Communications*, Seiten 149-165, June, 1988.
- [167] Schauer G. *Implementierung eines Schicht 2 Protokolls für ein LAN-Meßsystem*. Studienarbeit Nr. 1053, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1990.
- [168] Schmid, M.: *Entwurf und Implementierung von Meßapplikationen für ein verteiltes LAN-Meßsystem*. Diplomarbeit Nr. 1092, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1991.
- [169] Schwinn, B.: Parallelität und intelligentes Backtracking in logischen Programmen. *Informationstechnik - it*, 31. Jahrgang, Heft 4, Seiten 239-244, April 1989.
- [170] Seliger, P.: *Entwicklung einer graphischen Benutzeroberfläche für ein Monitoringsystem unter OSF/Motif*. Studienarbeit, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1992.
- [171] Schöllkopf, K.: *Entwicklung und Implementierung von LAN-Testsoftware für einen Ethernet LAN-Tester*. Studienarbeit Nr. 1101, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1991.
- [172] Schollenberger, W.: Monitoringsystem für ein verteiltes Meßsystem. *Ausgewählte Beiträge zum Workshop "Entwicklungstendenzen von Rechnernetzen"*, Band 2, Fakultät Informatik der Technischen Universität Dresden, Gaußig, Seiten 172-179, 18.-22. November 1991.

- [173] Schollenberger, W.: Netzmanagement. Beitrag zum Hochschulkolloquium, Universität Stuttgart, Institut für Nachrichtenvermittlung und Datenverarbeitung, 1990.
- [174] Schollenberger, W.; Rössler, G.; Bosch, M.: Network Management in Heterogeneous Networks for Factory Automation. *Proceedings Information Network and Data Communication (INDC)*, Lillehammer, Seiten 1A/2-1-13, Norway, 1990.
- [175] Schöneburg, E.; Hansen, N.; Gawelczyk, A.: Neuronale Netzwerke. Markt & Technik Verlag, Haar bei München, 1990.
- [176] Stümpfle, M.: *Entwurf und Implementierung eines Regelinterpreters für ein Diagnosesystem*. Diplomarbeit Nr. 1091, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1991.
- [177] Suppan-Borowka, J.: Das Manufacturing Automation Protocol MAP. *PIK, Praxis der Informationsverarbeitung und Kommunikation*, 10. Jahrgang, Nummer 3, Seiten 182-188, 1987.
- [178] Suppan-Borowka, J.: Ethernet Kontra Token Ring. *DATACOM*, 6. Jahrgang, Heft 2, Seiten 122-128, Februar 1989.
- [179] Sutter, M. T., Zeldin, P., E.: Designing Expert Systmes for Real-Time Diagnosis of Self-Correcting Networks. *IEEE Network*, Vol.2, No. 5, Seiten 43-51, September 1988.
- [180] Saylor, M., W.: Managing Phase V DECnet Newtworks: the Entity Model. *IEEE Network*, Vol. 2, No. 2, Seiten 30-36, 1988.
- [181] Tiribelli, P.: Trends and Progress in Network Operations in Europe. *IEEE NOMS 88, Conference Record*, New Orleans, Februar 1988.
- [182] Tow, D., M.: Network Management - Recent Advances and Future Trends. *IEEE Journal on Selected Areas in Communications*, Vol. 6, No. 4, Seiten 732-741, 1988.
- [183] Unbehauen, R.: Systemtheorie. Oldenbourg Verlag, München, Wien, 1983.
- [184] Vogt, H.-G.: *Feldbussystem für verteilte Sensoren*. Diplomarbeit Nr. 1103, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1991.
- [185] Wagner, T.: *Aufbau einer Wissenserwerbskomponente für ein modulares, verteiltes Diagnosesystem auf dem Betriebssystem UNIX*. Diplomarbeit Nr. 1097, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1991.
- [186] Watson, H., J.; Mann, R., I.: Expert Systems: Past, Present, and Future. *Journal of Information Systems Management*, Vol.5, No. 4, Seiten 39-46, 1988.
- [187] Weixler, M.: Distributed Measurement System for Protocols and Applications in ISO 8802/3 LANs, *International Federation for Information Processing (IFIP), Fourth Conference on Data Communication Systems and Their Performance*, Barcelona, Seiten 448-455, 20.-22. Juni 1990
- [188] Willett, M.; Martin, R., D.: LAN Management in an IBM Framework. *IEEE Network*, Vol.2, No. 2, Seiten 6-12, 1988.
- [189] Willetts, K.: Concert - an initiative towards open integrated network management. *Computer Communications*, Vol. 13, No. 9, Seiten 527-532, September 1990.
- [190] Winckler, A.: *Aufbau einer modularisierten Wissensbasis für ein Diagnose-Expertensystem für das IND Rechnernetz*. Studienarbeit Nr. 971, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1989.

- [191] Witt, K.: Netzmanagement - Problembestimmung und Lösungsansätze -. *Angewandte Informatik*, Heft 8, Seiten 323-328, 1986.
- [192] Yamahira, T.; Kiriha, Y.; Sakata, S.: Network Troubleshooting Experts System EXNETS. *NEC Research & Development*, No. 94, July 1989.
- [193] Yudkin, R., O.: On Testing Communication Networks. *IEEE Selected Areas in Communications*, Vol. 6, No. 5, Seiten 805-812, Juni 1988.
- [194] Yushida, M.; Namba, D.; Ikeda, S.; Baba, M.: Application of Knowledge Engineering to Computer System Diagnosis. *Review of the Electrical Communications Laboratories*, Vol. 37, No. 1, Seiten 9-14, Januar 1989.
- [195] Zhan, W.; Thanawastien, S.; Delcambre, L.: An Expert Workstation for Designing Rule-Based Network Management Systems. *IEEE Network*, Vol. 2, No. 5, Seiten 35-42, September 1988.
- [196] Zadeh, L., A.: Fuzzy Logic. *IEEE Computer*, Vol. 21, No. 4, Seiten 83-92, April 1988.
- [197] Zadeh, L., A.: Making computers think like people. *IEEE Spectrum*, Vol. 21, No. 8, Seiten 26-32, August 1984.
- [198] Zadeh, L., A.: The role of fuzzy logic in the management of uncertainty in expert systems. *Fuzzy Sets and Systems*, North-Holland, Vol. 11. Seiten 199-227, 1983.

