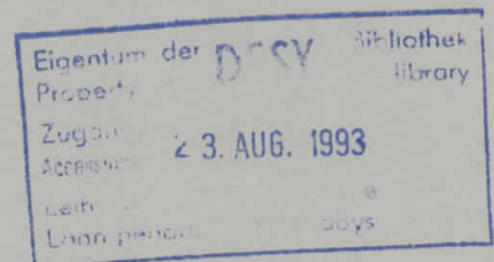


Interner Bericht
DESY F35D-93-05
Juli 1993

**Entwicklung des Prototypen
eines Echtzeit-Expertensystems
zur Diagnose von Fehlern in der ZEUS-Datennahme**

von

K. Ohrenberg



DESY behält sich alle Rechte für den Fall der Schutzrechtserteilung und für die wirtschaftliche Verwertung der in diesem Bericht enthaltenen Informationen vor.

DESY reserves all rights for commercial use of information included in this report, especially in case of filing application for or grant of patents.

**“Die Verantwortung für den Inhalt dieses
Internen Berichtes liegt ausschließlich beim Verfasser“**

**Entwicklung des Prototypen eines
Echtzeit-Expertensystems zur Diagnose von
Fehlern in der ZEUS-Datennahme**

Diplomarbeit am Fachbereich Physik der Universität Hamburg

vorgelegt von
Kars Ohrenberg ✓

Juli 1993

Inhaltsverzeichnis

Einleitung	1
1 Das ZEUS-Experiment	3
1.1 Der ZEUS-Detektor	3
1.2 Das ZEUS-Datenerfassungssystem	6
1.3 Die Experimentsteuerung	9
2 ZEX — Ein Expertensystem für ZEUS	13
2.1 Künstliche Intelligenz und Expertensysteme	13
2.2 Einsatzbereich und Aufgabenbeschreibung von ZEX	16
2.3 Problemanalyse	17
2.3.1 Das Überwachungssystem	18
2.3.2 Das Diagnosesystem	20
2.3.3 Das Kontrollsystem	20
2.4 Ein Entwurf für das Expertensystem ZEX	20
3 Entwurf und Implementation eines Prototypen: ZEX-P	23
3.1 Die Arbeitsumgebung	23
3.2 Die Systemanalyse	26
3.3 Die Implementation von ZEX-P	30
3.3.1 Die Online-Überwachungsdaten	30
3.3.2 Diskussion der verwendeten Methoden	31
3.3.3 Architektur und Arbeitsweise des Prototypen	36
3.4 Die Benutzerschnittstelle	42
4 Erste Betriebserfahrungen mit ZEX-P	45
4.1 Der Weg zum Online-System	45
4.2 Der Prototyp im Einsatz	48
4.2.1 Fallstudien	48
4.2.2 Eine Systemblockade	50
4.2.3 Eine komplizierte Situation und die Vorteile eines ES	53
4.3 Möglichkeiten und Grenzen des Prototypen	56
4.4 Der Prototyp als Studie für ZEX	59
4.4.1 Realisierbarkeit eines Expertensystems	59
4.4.2 Betrachtungen zur Implementation von ZEX	59
4.5 Ausblick	60
Literaturverzeichnis	62

Abbildungsverzeichnis

1.1	Der ZEUS-Detektor	4
1.2	Das Datenerfassungssystem des ZEUS-Detektors	7
1.3	Die Systemzustände des Datenerfassungssystems	11
2.1	Datenflußdiagramm des geplanten ZEUS-Expertensystems	19
2.2	Die Systemschnittstellen des geplanten Expertensystems	21
3.1	Die Funktionseinheiten des Eventbuilders	24
3.2	Das 2TP-VME-Modul als Komponentenschnittstelle	25
3.3	Das 2TP-VME-Modul als TLT-Schnittstelle	26
3.4	Ein Fehlverhalten einer Komponente	28
3.5	Ein Abnahmeproblem des Third Level Triggers	29
3.6	Die Methode der Cluster Zentren	33
3.7	Notation eines Automaten und ein erstes Beispiel	35
3.8	Gegenüberstellung von Automaten und regelbasierten Systemen	36
3.9	Datenflußdiagramm des Prototypen des ZEUS-Expertensystems	37
3.10	Der Hauptautomat des Überwachungs-/Diagnosemoduls	39
3.11	Die Arbeitsweise des Überwachungs-/Diagnosemoduls	40
3.12	Automat zur Vorhersage einer Systemblockade	41
3.13	Die Benutzerschnittstelle des Expertensystems	42
4.1	Ein kosmisches Ereignis	46
4.2	Ein Beam-Gas-Ereignis	47
4.3	Das Expertensystem vor dem Starten eines Runs	49
4.4	Die Expertensystembenutzerschnittstelle beim Starten eines Runs.	49
4.5	Ein Komponentengefährdungsverhalten und die Reaktion des Expertensystems	51
4.6	Ein Abnahmeproblem des TLT und die Reaktion des Expertensystems	52
4.7	Eine Systemblockade	53
4.8	Die ersten Anzeichen einer möglichen Blockade	54
4.9	Die Vorhersage der Blockade	54
4.10	Die Blockade	55
4.11	Die Ankündigung der Systemerholung	55
4.12	Die Systemerholung	56
4.13	Eine komplizierte Situation (Teil 1)	57
4.14	Eine komplizierte Situation (Teil 2)	58

Tabellenverzeichnis

1.1	Übersicht der verschiedenen Komponentensubsysteme des ZEUS-Experiments	8
2.1	Einsatzgebiete und Aufgabenbereiche von Expertensystemen	16
2.2	Die Aufgabenbereiche des ZEUS-Expertensystems	19
3.1	Die Offline-Überwachungsdaten	27
3.2	Die Online-Überwachungsdaten	32

Einleitung

Die vorliegende Diplomarbeit wurde im Rahmen des an der Hadronen-Elektronen-Ringanlage (HERA) des Deutschen Elektronen-Synchrotrons in Hamburg durchgeführten ZEUS-Experiments erstellt. Bei den heutigen in den Experimenten der Hochenergiephysik eingesetzten Universalnachweisgeräten wie dem ZEUS-Detektor handelt es sich um sehr komplexe Systeme. Sie dienen der Aufzeichnung der zu beobachtenden Teilchenreaktionen und müssen in der Lage sein, so unterschiedliche Parameter wie Teilchenspuren, -impulse, -energien usw. präzise zu bestimmen. Da hierfür unterschiedliche Meßverfahren angewandt werden, sind die Detektoren aus mehreren, auf die Bestimmung einzelner Parameter ausgelegter Komponenten aufgebaut. So verfügt der ZEUS-Detektor über Spurrkammern, Kalorimeter etc. und ist aus insgesamt siebzehn Subdetektoren aufgebaut. Jede dieser Komponenten erfordert eine auf ihre jeweilige Funktionsweise abgestimmte Ausleseelektronik, so daß sich ein sehr heterogener Aufbau des Datenerfassungssystems ergibt. Dieses hat neben der für spätere physikalische Analysen notwendigen Aufzeichnung der Kollisionsdaten die Aufgabe, eine Selektion der anfallenden Informationen vorzunehmen. Dies ist erforderlich, da einerseits zumeist nicht interessierende Ereignisse wie kosmische Schauer oder Beam-Gas-Reaktionen zu Signalen im Detektor führen und andererseits an der Eingangsseite eine, mit heutiger Technik nicht aufzeichnenbare Datenmenge von 5 TBytes/s anfällt. Aus diesem Grund erfolgt die Datennahme mittels eines parallel und asynchron arbeitenden, dreistufigen Triggersystems.

In diesem komplexen Datennahmesystem können so unterschiedliche Fehler wie der Ausfall einer Komponente, Kommunikations- oder Synchronisationsprobleme, Datenflußblockaden usw. auftreten und zu einer geringen Systemleistung oder auch Systemausfällen führen. Um eine bestmögliche Ausnutzung der HERA-Strahlmeßzeit zu gewährleisten, besteht die Aufgabe der Bedienermannschaft des ZEUS-Experiments darin, eine möglichst hohe Verfügbarkeit des Detektors zu gewährleisten, d. h. die beschriebenen Probleme schnellstmöglich zu erkennen und die Fehlerquelle zu beseitigen. Dafür steht den Operateuren aufgrund der Komplexität des Detektors und seines Datennahmesystems eine nahezu unüberschaubare Anzahl von Überwachungsinformationen zur Verfügung. Eine effektive Steuerung des Experiments erfordert eine Interpretation der vorliegenden Überwachungsdaten, wobei die Informationsfülle eine Selektion der vorliegenden Daten seitens der Operateure notwendig macht. Diese Auswahl erfordert Erfahrung und umfassende Kenntnisse der Systemparameter, jedoch ist das dafür benötigte Wissen auf die jeweiligen Systemexperten verteilt und steht daher nicht zentral und jederzeit zur Verfügung.

Um die geschilderten Probleme der Bediener bei der Steuerung des Experiments zu umgehen, bietet sich der Einsatz eines Expertensystems an. Bei diesen Systemen handelt es sich um eine aus dem Bereich der künstlichen Intelligenz stammende Anwendung, deren Ziel darin besteht, menschliche Experten zu simulieren. Diese Systeme verfügen über eine Wissensbasis, in der die Kenntnisse vieler Experten verankert werden sowie einen Mechanismus, um Schlußfolgerungen aus diesem Wissen abzuleiten. Ein derartiges System wurde erstmals im Rahmen

des am CERN durchgeführten ALEPH-Experiments zur Steuerung eines Hochenergiephysik-experiments erfolgreich eingesetzt. Zur Unterstützung der ZEUS-Bedienermannschaft wird im Rahmen des ZEUS-Experiments gegenwärtig ein Expertensystem, das ZEUS Experten System (ZEX) entwickelt. Dieses System wird zur Überwachung der Datennahme eingesetzt werden und anhand der zur Verfügung stehenden Überwachungsdaten Fehler im Datennahmesystem erkennen und Fehlerdiagnosen vornehmen. Diese Informationen werden der Bedienermannschaft bereitgestellt, so daß diese oder ein in das Expertensystem integriertes Kontrollsystem Befehlssequenzen zur Steuerung des Experiments einleiten können.

Im Rahmen dieser Diplomarbeit wurde ein Prototyp (ZEX-P) des angestrebten Expertensystems entwickelt und im April 1993 in Betrieb genommen. Über die Entwicklung, die Implementation, sowie die aus den ersten Betriebswochen gewonnenen Erfahrungen wird nachfolgend berichtet.

Kapitel 1

Das ZEUS-Experiment

Das ZEUS-Experiment ist eines der beiden großen Experimente der Hochenergiephysik, die zur Zeit am HERA-Speicherring des Deutschen Elektronen-Synchrotrons in Hamburg durchgeführt werden. Ziel der Experimente ist die Untersuchung der Grundbausteine der Materie und der zwischen ihnen herrschenden Wechselwirkungen.

Die bisher durchgeführten Experimente der Hochenergiephysik haben zu einem detaillierten Verständnis der Struktur der Materie beigetragen und zur Entwicklung einer sehr erfolgreichen Theorie, dem Standardmodell, geführt. Dieses Modell ermöglicht die Beschreibung nahezu aller bisher beobachteten Phänomene und erlaubt sehr präzise Vorhersagen wie die Existenz der W-Bosonen. Bei dem Standardmodell in seiner heutigen Form kann es sich allerdings nicht um eine endgültige, alles umfassende Theorie handeln. So läßt sich die Gravitation nicht zufriedenstellend in das Modell integrieren und eine Vielzahl von Naturkonstanten, wie beispielsweise die Teilchenmassen, folgen nicht direkt aus dieser Theorie, sondern müssen als äußere Parameter eingesetzt werden. Weiterhin konnten einige, dem Standardmodell zufolge existierende Grundbausteine, wie das Top Quark oder das Higgs Teilchen, noch nicht nachgewiesen werden, so daß weiterhin umfangreiche Untersuchungen zur Struktur der Materie erforderlich sind.

Um die notwendigen Experimente in bisher unerforschten Energiebereichen durchführen zu können, wurde beim DESY im Jahr 1984 mit dem Bau der weltweit ersten und einzigen Elektron-Proton-Speicherringanlage begonnen. Nach einer Bauphase von sechs Jahren konnte der Ring 1990 offiziell eingeweiht und am 19. Oktober 1991 die erste e-p-Kollision beobachtet werden. Der Beschleuniger bietet die Möglichkeit, die Streuung von Elektronen und Protonen bei einer Elektronenenergie von 30 GeV und einer Protonenenergie von 820 GeV zu untersuchen.

Zur Aufzeichnung der auftretenden e-p-Streureaktionen wurde in einer Gemeinschaftsarbeit von etwa 450 Wissenschaftlern aus 11 Ländern und 50 wissenschaftlichen Instituten der ZEUS-Detektor entwickelt und im April des Jahres 1992 in Betrieb genommen. Dieser Detektor, sein Datenerfassungssystem, sowie die Steuerung des Experiments werden im folgenden vorgestellt.

1.1 Der ZEUS-Detektor.

Zur physikalischen Interpretation der e-p-Streureaktionen ist eine präzise Vermessung der Bahnen, Impulse, Energien und weiterer Eigenschaften der bei den Kollisionen erzeugten Teilchen und Teilchenjets notwendig. Die Bestimmung dieser verschiedenartigsten Parameter erfordert unterschiedlichste Meßverfahren und Nachweisgeräte, so daß der ZEUS-Detektor aus insgesamt 17 Komponenten aufgebaut ist. Der Detektor erreicht damit bei Abmessungen von 12 m × 10 m × 19 m ein Gesamtgewicht von 3600 Tonnen. Die Anordnung der einzelnen Kom-

ponenten erfolgt im wesentlichen zwiebelschalenförmig, wobei die inneren Komponenten der Spur- und die äußeren der Energiebestimmung dienen. Diese Anordnung ist notwendig, da die Energiemessung zur Absorption der Teilchen führt.

Anhand der Abbildung 1.1 soll hier eine kurze Beschreibung der Detektorkomponenten erfolgen, wobei detaillierte Informationen [ZEU 93] zu entnehmen sind.

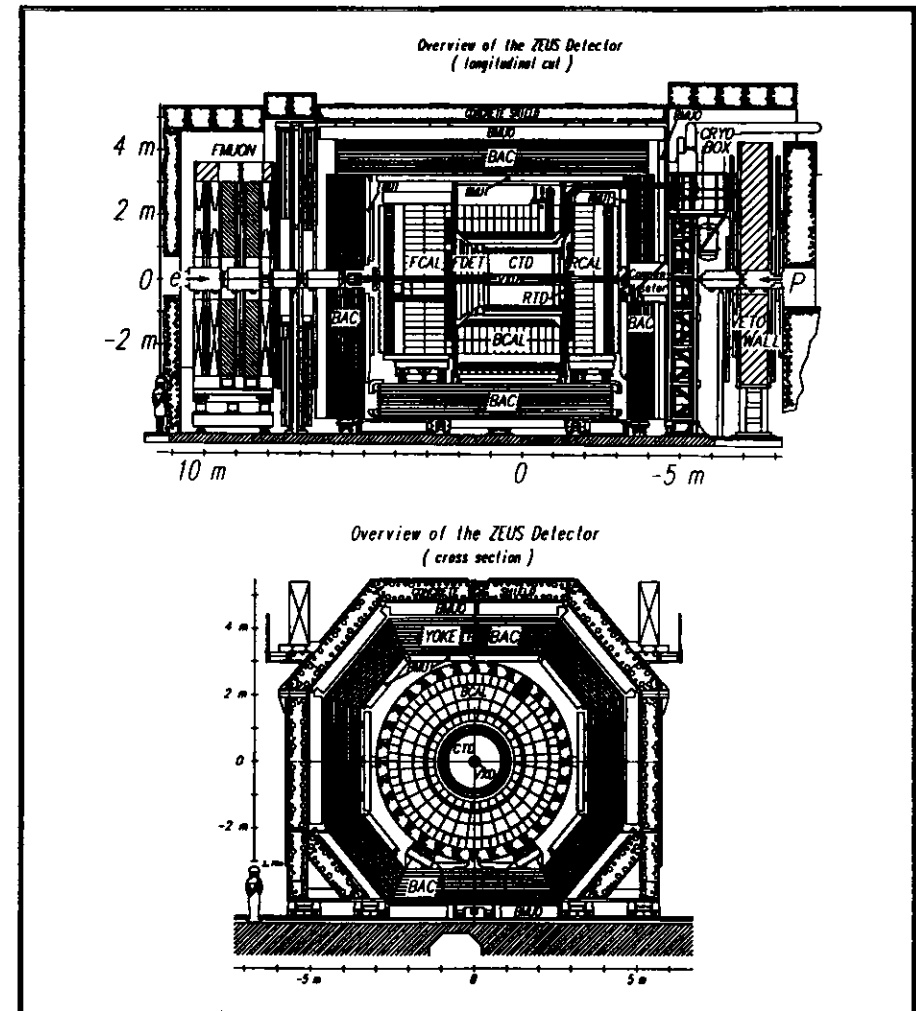


Abbildung 1.1: Der ZEUS-Detektor (Längs- und Querschnitt). Eine nähere Beschreibung der einzelnen Detektorelemente wird im Text gegeben.

- Das innerste, direkt am Strahlrohr befindliche Nachweisgerät bildet die Vertex Kammer

(VXD), eine kleine, zylindrische Zeit-Expansionskammer mit sehr guter Ortsauflösung im Bereich von $35 - 70 \mu\text{m}$. Sie dient der exakten Vermessung des Elektron-Proton-Wechselwirkungspunktes und der Aufzeichnung kurzlebiger Teilchen.

- Die Vertex Kammer wird von einer zylindrischen Driftkammer, der zentralen Spurkammer (Central Tracking Detector, CTD) umschlossen. Diese Kammer verfügt über eine Ortsauflösung, die im Bereich von $120 - 130 \mu\text{m}$ in der $r-\phi$ -Ebene und $1,0 - 1,4 \text{ mm}$ in axialer Richtung liegt. Damit wird bei einem Magnetfeld von $1,8 \text{ T}$ eine Impulsauflösung von $\frac{\sigma_p}{p} = 0,2\text{p}\%$ erreicht.
- Die zentrale Spurkammer und der Vertexdetektor werden von einer supraleitenden Spule (COIL) umgeben, deren nominales Magnetfeld von $1,8 \text{ T}$ in Verbindung mit den Spurkammern zur Impulsbestimmung der Reaktionsprodukte benutzt wird. Der Einfluß dieser Spule auf den Speicherring wird durch eine weitere, supraleitende Spule (COMPENSATOR) ausgeglichen. Diese Spule erzeugt ein Magnetfeld von 5 T und befindet sich in Elektronenrichtung hinter dem RCAL.
- Entlang der Strahlachse schließen sich in Elektronenrichtung die Rückwärts- und in Protonenrichtung die Vorwärtsspurkammern an (RTD, Rear Tracking Detector; FTD, Forward Tracking Detector). Der RTD besteht aus einer, der FTD aus drei ebenen Driftkammern die der Abdeckung der von den zentralen Spurkammern nicht erfaßten Winkelbereiche dienen. Diese Detektoren erreichen Ortsauflösungen von $120 \mu\text{m}$ und ermöglichen eine Impulsauflösung von $\frac{\sigma_p}{p} = 1\text{p}\%$ bei einem Winkel von 140 mrad .
- Die Freiräume zwischen den drei Driftkammern des FTD dienen der Aufnahme der beiden Übergangstrahlungsdetektoren (TRD, Transition Radiation Detector). Diese Detektoren dienen insbesondere in Jets der Unterscheidung von Hadronen und Elektronen.
- Zur Energiebestimmung der Reaktionsprodukte werden die bisher angeführten Komponenten, welche den zentralen Bereich des Detektors darstellen, von einem Uran-Szintillator-Kalorimeter umgeben. Dieses Kalorimeter erreicht eine Energieauflösung von $\frac{\sigma_E}{E} = \frac{10}{\sqrt{E}}\%$ für Elektronen und $\frac{\sigma_E}{E} = \frac{35}{\sqrt{E}}\%$ für Hadronen, für Jets eine Ortsauflösung von 10 mrad und eine Raumwinkelabdeckung von etwa $99,8\%$ in Protonen- und $99,5\%$ in Elektronenrichtung. Das Kalorimeter ist aus insgesamt 80 Modulen aufgebaut, die sich in drei Bereiche, das FCAL (Forward Calorimeter), das RCAL (Rear Calorimeter) und das BCAL (Barrel Calorimeter) gliedern. Jedes Modul besitzt eine Lagenstruktur aus Platten abgereicherten Urans und Schichten Szintillatormaterials.
- Zur besseren Unterscheidung von Elektronen und Hadronen sind hinter der dritten und siebten Szintillatorschicht der FCAL-Module, sowie hinter der vierten Szintillatorschicht der BCAL- und RCAL-Module, die Siliziumdioden des Hadron Elektron Separators (HES, auf den Abbildungen nicht dargestellt) vorgesehen. Diese Komponente befindet sich gegenwärtig im Aufbau, wobei bisher etwa $1/3$ der FCAL Fläche mit Dioden bestückt worden ist.
- Das den gesamten inneren Detektor und das Uran-Szintillator-Kalorimeter umschließende Backing Calorimeter (BAC) vervollständigt die Energiemessung und dient der Restenergiebestimmung von nicht vollständig im Kalorimeter erfaßten hadronischen Schauern. Das

BAC wird aus den sechs Schichten des Eisenjochs (YOKE) und den in den Zwischenräumen befindlichen Proportionalzählrohren gebildet und erreicht eine Energieauflösung von $\frac{\sigma_E}{E} = \frac{110}{\sqrt{E}}\%$, sowie eine Ortsauflösung von etwa 1 mm .

- Innerhalb und außerhalb des Eisenjochs befinden sich weitere Spurkammern, die zum Nachweis von Myonen eingesetzt werden (BMUO, BMUI, RMUO, RMUI, Barrel/Rear Muon Outer/Inner Detector). Da das Eisen des Jochs durch Kupferspulen auf $1,6 \text{ Tesla}$ erregt wird, ist mit diesen Kammern eine Impulsbestimmung der Myonen möglich. Die Auflösung beträgt $\frac{\sigma_p}{p} < 30\%$ bei einer Myonenenergie von 20 GeV .
- In Protonenrichtung schließt sich außerhalb des Eisenjochs das Vorwärts Myon Spektrometer an (FMUO). Innerhalb des Eisenjochs besteht es wie BMUI und RMUI aus einer inneren Driftkammer, außerhalb aber aus einem Driftkammersystem mit zwischengelagerten, torusförmigen und magnetisierten Eisenplatten. Das Spektrometer erreicht bei Energien von 100 GeV eine Auflösung von $\frac{\sigma_p}{p} < 23\%$.
- In Elektronenrichtung schließt sich außerhalb des Eisenjochs die Vetowand (VETO) an. Diese, mit Szintillatorschichten auf beiden Seiten bestückte Eisenwand dient der Abschirmung des zentralen Detektorbereiches vor Teilchen aus dem Strahl-Halo.

Neben den bisher erwähnten Komponenten existieren noch zwei weitere Detektorkomponenten die sich nicht mehr in der Nähe des Wechselwirkungspunktes befinden und daher auf den Abbildungen nicht dargestellt sind. Es handelt sich dabei um:

- Das Leading Proton Spectrometer (LPS). Es dient dem Nachweis von Protonen, die unter sehr kleinen Winkeln in Vorwärtsrichtung gestreut werden. Es besteht aus sechs Detektorstationen, die im Protonenspeicherring über eine Länge von $40 - 90 \text{ m}$ verteilt sind.
- Den Luminositätsmonitor (LUMI). Er dient der Bestimmung der Luminosität des Speicherrings. Dazu werden die Photonen / Elektronen der Elektron-Proton-Bremsstrahlung in je einem, vom Wechselwirkungspunkt $108 \text{ m} / 36 \text{ m}$ entferntem Kalorimeter nachgewiesen.

1.2 Das ZEUS-Datenerfassungssystem

Neben seiner Hauptaufgabe, der Auslese des ZEUS-Detektors und der Aufzeichnung der registrierten e-p-Kollisionen, erfüllt das Datenerfassungssystem weitere Aufgaben wie die Datenfilterung und Experimentkontrolle. Zudem dient es der Überwachung und Kontrolle des Datenflusses, der Speicherung und Verwaltung von Kalibrationskonstanten, sowie der Überwachung der Hard- und Softwareperformance. Auf die an das Datennahmesystem zu stellenden Anforderungen und dessen Aufbau wird nachfolgend näher eingegangen.

Beim Betrieb des HERA-Speicherrings kann alle 96 ns eine vom Detektor zu erfassende Elektron-Proton-Reaktion auftreten, so daß die rund 250.000 Datenkanäle des ZEUS-Detektors mit einer Rate von etwa 10 MHz ausgelesen werden müssen. Die Aufzeichnung aller möglichen Kollisionen erfordert bei einer pro Detektorauslese anfallenden Rohdatenmenge von 500 MBytes , eine vom Datenerfassungssystem zu erbringende Verarbeitungsleistung von 5 TBytes/s . Da die Aufzeichnung und Analyse einer derartigen Datenmenge mit heutiger Technik nicht möglich ist, muß eine Selektion der Daten vorgenommen werden. Dies ist möglich, da es neben den

zu beobachtenden e-p-Kollisionen wesentlich häufiger zu kaum interessierenden Untergrundeignissen wie Teilchenverlusten aus dem Strahl, Reaktionen mit Restgasatomen im Strahlrohr oder kosmischen Schauern kommt.

Zur Handhabung der extrem hohen Datenraten und der notwendigen Datenreduktion und -selektion wurde das in Abbildung 1.2 dargestellte, dreistufige, parallel und asynchron operierende Datennahmesystem entwickelt. Die Arbeitsweise dieses Systems wird im folgenden kurz vorgestellt, wobei Details, insbesondere zu den verwendeten Triggeralgorithmen, [Smi 90, ZEU 93] zu entnehmen sind.

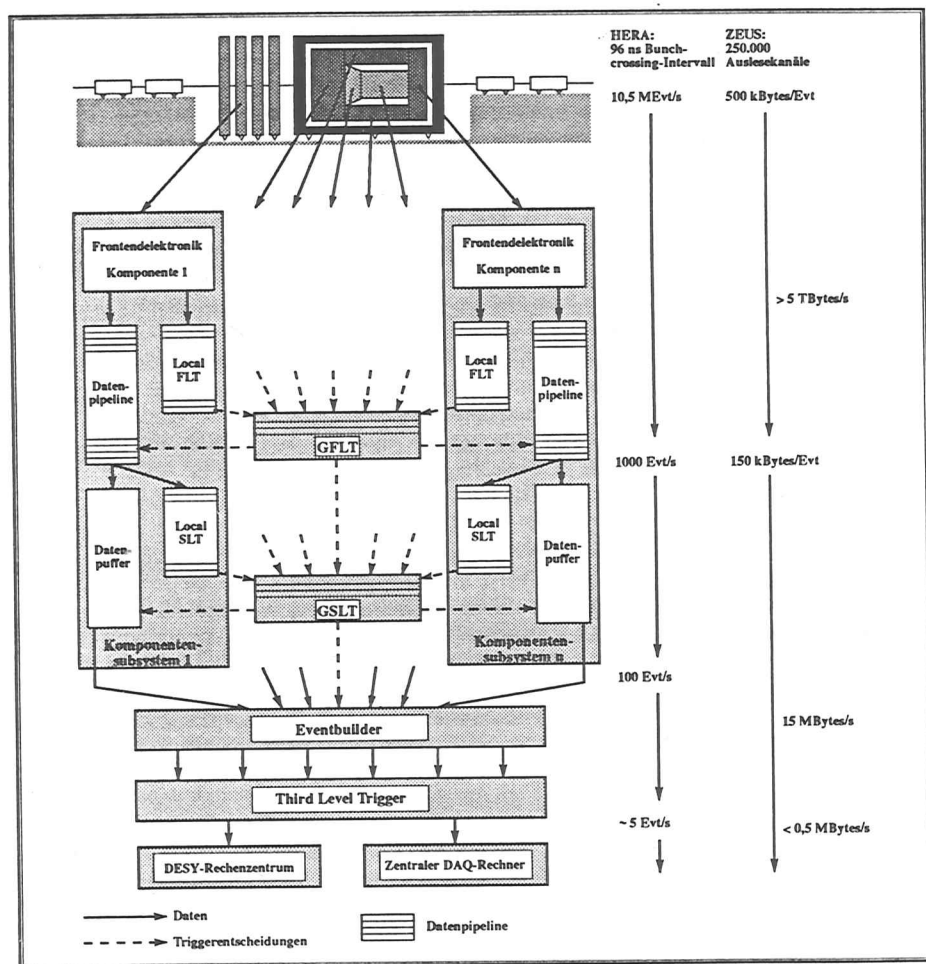


Abbildung 1.2: Das Datenerfassungssystem des ZEUS-Detektors. (Nähere Erläuterungen im Text).

Die erste Stufe des Datennahmesystems bilden die parallel und unabhängig voneinander operierenden Komponentensubsysteme. Diese beinhalten die zum Betrieb und zur Datenauslese der jeweiligen Komponente erforderliche Elektronik. Die Komponenten wurden entsprechend der internationalen Zusammensetzung der ZEUS-Kollaboration von unterschiedlichen Instituten aufgebaut und werden auch von diesen betrieben. Da jede Komponente eine auf ihre jeweilige Arbeits- und Funktionsweise abgestimmte Computerhard- und Software erfordert, welche ebenfalls von den jeweiligen Instituten entwickelt wurde, ergibt sich auf dieser Ebene ein sehr heterogener Aufbau des Datennahmesystems. Eine Auflistung der von den jeweiligen Komponentensubsystemen auszulesenden Datenkanäle und der pro Auslesezyklus typischerweise anfallenden Datenmengen findet sich in Tabelle 1.1. Die Auslesekanäle werden vom Komponentensubsystem alle 96 ns ausgewertet und die Informationen je nach Beschaffenheit der Ausleseelektronik analog oder bereits digitalisiert in einer 52 Einträge umfassenden Datenpipeline abgelegt.

Komponente	Anzahl der Auslesekanäle	Datenlänge [kBytes]	
Central Tracking Detector	CTD	4.608	10 – 30
Forward/Rear Tracking Detector	FRTD	5.778	10 – 15
Barrel Calorimeter	BCAL	5.184	5 – 20
Forward Calorimeter	FCAL	4.344	5 – 20
Rear Calorimeter	RCAL	2.336	5 – 10
Transition Radiation Detector	TRD	2.472	5 – 10
Hadron Electron Separator	HES	37.304	5 – 10
Backing Calorimeter	BAC	≈ 40.000	2
Vertex Detektor	VXD	832	2
Beamline	BEAM	31	1
Barrel/Rear Muon Detector	BMUO	84.336	0,9
Forward Muon Detector	FMUO	18.948	0,5
Leading Proton Spectrometer	LPS	52.000	0,2
Luminosity Monitor	LUMI		0,2
Vetowall	VETO		0,01
Fast Clear	FCLR		1,2
Global Second Level Trigger	GSLT		10 – 20
Σ		258.173	62 – 142

Tabelle 1.1: Übersicht über die Komponentensubsysteme des ZEUS-Experiments. (Nach [Sch 92])

Eine erste Selektion der Daten erfolgt durch die erste Triggerstufe, den First Level Trigger (FLT). Dieser hat innerhalb eines Zeitraums von $52 \cdot 96 \text{ ns} \approx 5 \mu\text{s}$ über die Verwertbarkeit der Daten zu entscheiden. Der rein auf Hardware basierende FLT ist in der Lage, einen Datenstrom von $10,5 \text{ GByte/s}$ zu verarbeiten, was bedeutet, daß er etwa 1/500 der anfallenden Datenmenge analysieren kann. Dies ist jedoch ausreichend, da für die vom FLT durchgeführten ersten, groben Analysen eine Auswertung der gesamten bei einer Detektorauslese anfallenden 500 kBytes nicht erforderlich ist. Hier ist es ausreichend, einige wenige Informationen wie beispielsweise

die gesamte im Kalorimeter deponierte Energie zu betrachten. Dazu führen zunächst lokale Triggersysteme der Komponenten innerhalb von $2\ \mu\text{s}$ die Analyse auf Teilmenge der zu einer Detektorauslese gehörigen Daten durch und übermitteln ihre Ergebnisse der globalen ersten Triggerstufe, dem Global First Level Trigger (GFLT). Beim Vorliegen aller lokalen Triggerinformationen trifft der GFLT innerhalb der verbleibenden $3\ \mu\text{s}$ eine Entscheidung über die Verwertbarkeit der Daten und teilt diese den Komponentensubsystemen mit. Die Rate der positiven GFLT Entscheidungen ist auf maximal $1000\ \text{Hz}$ festgelegt, wodurch sich die Anzahl der pro Sekunde zu verarbeitenden Ereignisse bereits um einen Faktor 10.000 reduziert hat.

Die vom GFLT akzeptierten Daten werden von den Komponentensubsystemen aus der $5\ \mu\text{s}$ -Datenpipeline in einen Datenpuffer transferiert, wo sie bis zur weiteren Verarbeitung durch eine zweite Triggerstufe gespeichert werden. Bei diesem Transfer erfolgt eine Komprimierung der Daten, wodurch sich die pro Detektorauslese anfallende Datenmenge auf etwa $150\ \text{MBytes}$ reduziert. Die zweite Filterstufe arbeitet entsprechend dem FLT, d. h. auch hier werden zunächst lokale Triggersysteme aktiv und führen im Vergleich zum FLT umfassendere Analysen der ihnen zur Verfügung stehenden Daten durch. Die resultierenden Ergebnisse werden abermals einer globalen Triggerstufe, dem Global Second Level Trigger (GSLT) zur Verfügung gestellt. Dieser trifft nach spätestens $3\ \text{ms}$ eine Entscheidung über die weitere Verwendung der jeweiligen Daten und teilt seine Auswahl dem Eventbuilder mit, der die Information an die Komponentensubsysteme weiterleitet. Der GSLT verwirft etwa 90 % der vom GFLT akzeptierten Daten, so daß nach Durchlaufen des GSLTs ein Datenstrom von etwa $15\ \text{MBytes/s}$ verbleibt.

Bis zu diesem Zeitpunkt wurden die von den Komponentensubsystemen registrierten Daten parallel und unabhängig voneinander verarbeitet, d. h. daß die zu einer Detektorauslese gehörigen Daten noch nicht in einer gemeinsamen Struktur, dem sogenannten Ereignis oder Event, zusammengefaßt sind. Dies ist Aufgabe der nächsten Verarbeitungsstufe, des Eventbuilders (EVB). Den vom GSLT akzeptierten Komponentendaten, den Ereignisteildaten, wird eine sogenannte GSLT Entscheidungsnummer zugewiesen, anhand derer es dem Eventbuilder möglich ist, die zusammengehörigen Ereignisteildaten zu einem Event zusammenzufassen und die vollständigen Ereignisse an die dritte Triggerstufe, den Third Level Trigger (TLT) weiterzureichen.

Bei dem TLT handelt es sich um eine über eine Rechenleistung von mehr als $1000\ \text{MIPS}$ verfügende Computerfarm aus 36 Workstations, die in Zweigen zu je sechs Workstations organisiert und dementsprechend über sechs Schnittstellen mit dem Eventbuilder verbunden ist. Jedes vom Eventbuilder erhaltene Ereignis wird einer, über freie Rechenkapazität verfügenden Workstation zugeleitet, woraufhin diese eine komplette Rekonstruktion und umfangreiche Analyse des Events durchführt. Auf der Basis dieser detaillierten Auswertungen wird eine letzte Entscheidung über die Verwertbarkeit des jeweiligen Ereignisses getroffen, wobei etwa fünf Events pro Sekunde akzeptiert werden.

Die vom TLT nicht verworfenen Daten werden über fiberoptische Leitungen zum zentralen Massenspeicher, einem IBM-Großrechner im DESY-Rechenzentrum transferiert und dort für spätere physikalische Analysen auf Magnetbändern abgelegt. Die Aufzeichnungsrate auf Seiten der Großrechenanlage ist begrenzt und liegt bei maximal $1,6\ \text{MBytes/s}$, wobei während der Datennahme typischerweise $0,5\ \text{MBytes/s}$ totzeitfrei transferiert werden können.

1.3 Die Experimentsteuerung

Die in Abschnitt 1.2 angesprochenen Komponenten des Datenerfassungssystems sind über ein Local Area Network (LAN) mit einem zentralen Experimentrechner verbunden. Die Steue-

rung des Experiments und der Datennahme (der zwischen dem Beginn und dem Ende einer Datennahme liegende Zeitraum wird dabei als Run bezeichnet) erfolgt über einen auf diesem Computer laufenden Prozeß, Run Control (RC). Dieses Programm gestattet es, die Datennahme zu initialisieren, zu starten, zu beenden, zu pausieren sowie abzubrechen und die für jeden Run erforderliche Runkonfiguration und Triggerbedingung festzulegen. Diese werden den Systemkomponenten mit dem Aktivierungsbefehl mitgeteilt und in einem sogenannten Begin of Run Event gespeichert.

Unter der Runkonfiguration ist dabei die Festlegung der am Run beteiligten Detektorkomponenten zu verstehen. Einzelne Komponenten können dabei sowohl aus technischen als auch physikalischen Gründen von der Datennahme ausgeschlossen werden.

Unter der Triggerkonfiguration ist die Auswahl der im vorherigen Abschnitt erwähnten Triggeralgorithmen und -bedingungen zu verstehen. Neben der Möglichkeit, durch eine gezielte Auswahl der Triggerbedingungen die üblicherweise verworfenen Daten wie kosmische Schauer oder Beam-Gas-Reaktionen registrieren zu können, ist vor allem eine Anpassung der Triggerbedingungen an die Strahlqualität erforderlich. Dies ist unbedingt notwendig, da eine bessere Strahlqualität und somit höhere e-p-Kollisionsrate bei konstanten Triggerbedingungen zu einer höheren Rate akzeptierter Ereignisse führen würde, die maximale Datenrate aber durch die begrenzte Aufzeichnungsrate des Massenspeichers limitiert ist.

Bei den zur Steuerung des Experiments zur Verfügung stehenden Befehlen handelt es sich um:

- Setup
Den Komponenten wird die Run- und Triggerkonfiguration mitgeteilt und entsprechende Initialisierungen durchgeführt.
- Activate
Die Komponenten werden veranlaßt, ihre Daten für das Begin of Run Event zu senden, die Trigger werden aktiviert und die Datennahme gestartet.
- Pause
Die Datennahme wird unterbrochen, wobei alle Konfigurationen erhalten bleiben.
- Resume
Die Datennahme wird mit der vor der Pausierung gültigen Konfiguration fortgesetzt.
- End
Die Komponenten werden veranlaßt, ihren Daten für das sogenannte End of Run Event zu senden, die Trigger werden deaktiviert und die Datennahme beendet.
- Abort
Die Datennahme wird abgebrochen, ohne den Run korrekt zu beenden. Dieses Kommando kann jederzeit eingeleitet werden.

Die Abfolge dieser Kommandos, sowie die Reihenfolge, in der diese Befehle an die Komponentensubsysteme verteilt und von diesen akzeptiert werden müssen, wird als Run Sequencing bezeichnet und ist in [You 91, ZEU 93] festgelegt.

Mit diesen Befehlen wird das Datenerfassungssystem (DAQ-System, Data Acquisition) in einen definierten Zustand, den DAQ-Status versetzt. Dabei sind die folgenden Zustände zu unterscheiden:

- **SETUP_IDLE**

Die Konfigurations- und Kalibrationsprozeduren sind für jede am Run beteiligte Komponente durchgeführt worden und die Datennahme kann beginnen. Die zur Aktivierung der Datennahme erforderlichen Startprozeduren (z. B. die Bildung eines Begin of Run Events) sind noch nicht ausgeführt worden. Eine Änderung der Hardwarekonfiguration ist nicht mehr möglich.

- **ACTIVE**

Alle Startprozeduren sind beendet und die Datennahme läuft. Eine Änderung der Runparameter ist in diesem Zustand nicht mehr möglich.

- **END_IDLE**

Alle zum Beenden eines Runs notwendigen Prozeduren wie die Bildung eines End of Run Events sind durchgeführt und die Datennahme ist beendet worden.

- **PAUSED**

Die Datennahme während des ACTIVE Zustandes ist durch eine Abschaltung der Trigger unterbrochen. Nur geringfügige Änderungen der Systemparameter sind erlaubt.

Abbildung 1.3 verdeutlicht nochmals das bisher Beschriebene und stellt die Systemzustände sowie die jeweils möglichen Befehle dar.

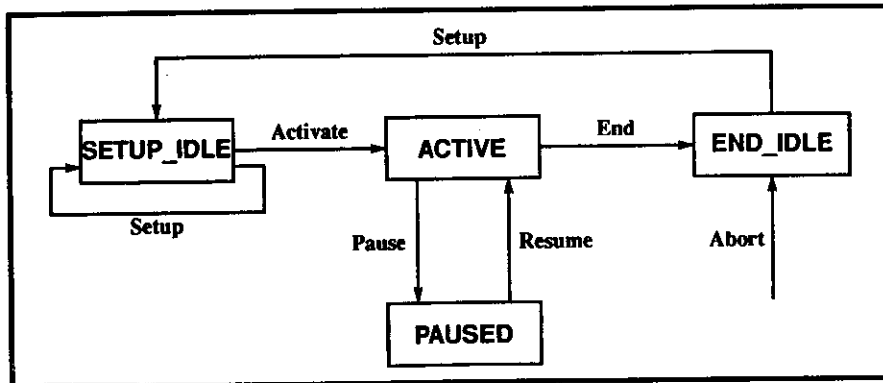


Abbildung 1.3: Die Systemzustände des Datenerfassungssystems und die möglichen Steuerbefehle. (Aus [ZEU 93])

Neben der Kontrolle des Datenerfassungssystems ist eine Kontrolle unterschiedlichster Detektorparameter wie Hochspannungen, Temperaturen, Gasdrücken und -zusammensetzungen, Magnetfeldstärken usw. erforderlich. Zur Überwachung und Steuerung dieser Parameter steht ein weiteres Kontrollprogramm, Slow Control zur Verfügung.

Kapitel 2

ZEX — Ein Expertensystem für ZEUS

Um die vom HERA-Beschleuniger für das ZEUS-Experiment zur Verfügung gestellte Strahlmeßzeit möglichst effizient ausnutzen zu können, wird eine hohe Systemverfügbarkeit des ZEUS-Detektors angestrebt. Aufgrund der im komplexen Datennahmesystem auftretenden Fehler und den daraus resultierenden Datenverlusten und Systemausfällen ist dieses Ziel nur bedingt zu erreichen. Zur Unterstützung der Bedienermannschaft bei der Überwachung des Datenerfassungssystems und der schnellstmöglichen Lokalisierung und Beseitigung einer Fehlerquelle wird gegenwärtig ein Expertensystem entwickelt [Beh92b, Beh 93].

2.1 Künstliche Intelligenz und Expertensysteme

Seit dem Beginn der Entwicklung moderner Computer besteht der Wunsch, diese Systeme mit 'Wissen' und 'Intelligenz' auszustatten. Diesem Ziel hat sich ein eigenständiger Zweig der Computerwissenschaft gewidmet, der Bereich der künstlichen Intelligenz (KI). Die Entwicklung und Realisation von Systemen mit künstlicher Intelligenz erfordert die Speicherung und Darstellung von Wissen. Zu diesem Zweck wurden eine Reihe unterschiedlichster Ansätze verfolgt und entwickelt, deren wichtigste nachfolgend kurz vorgestellt werden.

- **Heuristik**

Bei dieser Methode wird das Wissen in Form von Entscheidungsbäumen dargestellt und die Auswertung der Kenntnisse, d. h. eine Bestimmung des bestmöglichen Weges in diesen Bäumen, erfolgt über Suchalgorithmen. Die Stärke dieses Ansatzes liegt darin, daß er die Umsetzung praktisch jeglichen Wissens ermöglicht, jedoch einen sehr hohen Speicherbedarf (bei fest vorgegebenen Bäumen) oder großen Rechenaufwand (bei komplexen Problemen, z. B. dem Schachspiel, müssen die Bäume temporär erzeugt werden) erfordert, so daß in der Praxis schnell Grenzen erreicht werden.

- **Logik**

Dieser Ansatz beruht auf einer rein logischen Darstellung des verfügbaren Wissens. Dabei werden beispielsweise logische Verknüpfungen der Form $a < b$, $b < c$ aufgestellt, und Schlußfolgerungen wie $a < c$ gezogen. Diese Darstellung eignet sich insbesondere zur Überprüfung mathematischer Aussagen und Auswertung logischer Ausdrücke. Nachteilig ist die schwierige Handhabung der zugrundeliegenden Programmiersprachen (z. B. PROLOG).

- **Semantische Netzwerke**

Hier wird Wissen in Form von Objekten und ihnen zugewiesenen Eigenschaften repräsentiert. Die Objekte können entsprechend ihres Abstraktionsgrades hierarchisch in Klassen gegliedert werden, wobei die hierarchisch niedrigeren (die weniger abstrakten) ihre Eigenschaften von übergeordneten Objekten 'erben' können. Die konkreteren Objekte besitzen daher implizit alle Eigenschaften der ihnen übergeordneten Objekte, so daß die damit verbundenen Eigenschaften nicht explizit angegeben werden müssen, aber trotzdem zur Verfügung stehen¹. Diese Form der Wissensdarstellung bietet sich vor allem bei der Implementation komplexer Zusammenhänge mit vielschichtigen Hierarchien und Multirelationen an, nachteilig ist jedoch der hohe Speicherbedarf.

- **Symbolisch**

Dieser Ansatz beruht auf einer rein symbolischen Darstellung des bekannten Wissens, d. h. es existiert keine Trennung zwischen Prozeduren und Daten. Der Programmablauf erfordert dann lediglich Manipulationen und Ersetzungen der Symbole, wodurch dieser Ansatz bei entsprechend spezialisierter Hardware sehr effizient ist. Das Problem stellt die schwierige Handhabung der entsprechenden Programmiersprachen (z. B. LISP) dar.

- **Regelbasiert**

Die Wissensrepräsentation erfolgt durch in der Form *if ... then ...* aufgestellte Regeln. Der große Vorteil dieser Methode liegt in der Nähe zur menschlichen Denkweise, so daß diese Darstellung sehr intuitiv ist. Der Nachteil liegt in der schweren Programmierbarkeit, da in den verwendeten Programmiersprachen (z. B. OPS 5) keine Kontrollmöglichkeiten des Programmablaufes seitens des Programmierers vorgesehen sind.

- **Neuronale Netzwerke**

Diese Disziplin hat sich biologische Systeme als Vorbild genommen, und es wird versucht, das Zusammenwirken einzelner Nervenzellen zu simulieren. Jede dieser Zellen verfügt über Verbindungen zu Nachbarzellen, wobei die 'Wichtigkeit' der Verbindung durch entsprechende Bewertungsfaktoren ausgedrückt wird. Diese Faktoren ändern sich dynamisch und passen sich entsprechenden Vorgaben an, d. h. ein derartiges System kann lernen. Dies stellt auch den großen Vorteil dieses Ansatzes dar, das automatische Vorhandensein eines Lernmechanismus. Der große Nachteil liegt darin, daß es sich bei diesen Systemen um eine 'Black Box' handelt, so daß schwer nachvollziehbar ist, was wirklich geschieht. Aus diesem Grund sind die neuronalen Netzwerke nur für spezielle Problemlösungen einsetzbar.

- **Genetische Algorithmen**

Hier beruhen die zur Handhabung des Wissens erforderlichen Suchalgorithmen auf dem Darwinistischen Evolutionsprinzip. Die möglichen Lösungen werden durch 'Chromosomen' dargestellt, die mutieren und ihre Eigenschaften vererben können. Entsprechend dem biologischen Prinzip kann nur eine begrenzte Anzahl von Chromosomen 'überleben', so daß eine Selektion stattfindet und nur die 'besten' weiter existieren können. Für diesen sehr jungen und noch exotischen Ansatz gelten die selben Vor- und Nachteile wie bei den neuronalen Netzen.

- **Mustererkennung**

Hier wird das Wissen in Form bekannter Muster abgelegt. Die zugrundeliegenden Ver-

¹Dieses Prinzip bildet die Grundlage des sich zunehmender Beliebtheit erfreuenden objektorientierten Programmieransatzes.

fahren, um diese Muster in unbekanntem Daten wiederzuerkennen, sind der Mathematik entliehen (Vektoren, Statistik, Automatentheorie, Näherungsverfahren, ...) und daher sehr gut erforscht, entwickelt und etabliert. Ihr Nachteil ist der sehr eingeschränkte Einsatzbereich.

Die angesprochenen Methoden der künstlichen Intelligenz bieten sich insbesondere zur Analyse komplexer Probleme an, da in diesen Fällen die wissensbasierten Lösungen gegenüber den klassischen, datenbasierten Lösungen die ökonomischere und oftmals auch einzig realisierbare Möglichkeit darstellen. Typische Anwendungsbereiche der künstlichen Intelligenz finden sich in den Bereichen der Sprachanalyse, Robotik, CIM (Computer Integrated Manufacturing) oder Bildanalyse.

Einen weiteren Zweig der künstlichen Intelligenz bilden die Expertensysteme², deren Ziel die Simulation menschlicher Experten darstellt und die dadurch gekennzeichnet sind, daß sie über eine Wissensbasis und einen Schlußfolgerungsmechanismus zur Handhabung der Kenntnisse verfügen. In Expertensystemen kommen die angesprochenen Methoden der künstlichen Intelligenz zur Anwendung, insbesondere die Entscheidungsbäume, die regelbasierten Systeme und die Automaten, die auch als FSMs (Finite State Machine) bezeichnet werden.

Entsprechend seiner Aufgabe, der Simulation menschlicher Experten, läßt sich ein Expertensystem in fünf Komponenten aufteilen:

- Die Wissensbasis enthält in irgendeiner Form die zur Problemanalyse im jeweiligen Aufgabenbereich notwendigen Regeln und Fakten.
- Das Schlußfolgerungsmodul ist der Teil eines Expertensystems, der die Simulation des menschlichen Problemlösungsverhaltens darstellt. Der Schlußfolgerungsmechanismus wendet die in der Wissensbasis abgelegten Regeln auf vorhandene Fakten an. Dabei wird festgelegt, welche und vor allem in welcher Reihenfolge die Regeln angewandt werden. Unterschiedliche Probleme erfordern unterschiedliche Schlußfolgerungsmechanismen, so daß diese dem jeweiligen Problem angepaßt werden müssen.
- Das Lernmodul dient der Erweiterung der Wissensbasis durch Eingaben eines Experten oder des Systementwicklers, sowie durch Erfahrung während des Systembetriebes.
- Das Erklärungsmodul dient zur Erläuterung der Entscheidungen des Systems. Es stellt dem Benutzer die Schlußfolgerungen dar und gibt ihm die Möglichkeit, die Resultate nachzuvollziehen.
- Die Benutzerschnittstelle schließlich dient der Kommunikation mit dem Bediener. Sie bietet die Möglichkeit, das System zu steuern, und stellt die Ergebnisse in aufbereiteter Form dar.

Expertensysteme haben heutzutage ein weites Einsatzspektrum erreicht, von dessen Bandbreite die Tabelle 2.1 einen Eindruck vermittelt.

²Eine allgemeine Einführung in die Anwendungen und zugrundeliegenden Techniken von Expertensystemen findet sich in [Neb 88].

	Industrie	Öffentlicher Bereich	Bank/Versicherung
Überwachung und Kontrolle	Prozeßüberwachung, Prozeßkontrolle	Überwachung großer Netzwerke (Gas-, Wasserleitungen, ...)	Trendbeobachtung
Diagnose	Wartung	Medizinische Diagnose	Kreditvergabe, Risikoanalysen
Kontrolle	Prozeßkontrolle	Netzwerkkontrolle, Verkehrsleitsysteme	
Planung	Projektplanung, Schaltkreisentwurf	Planung z. B. von Versorgungsleitungen	Investitionsplanung
Simulation	Flugsimulator	Verkehrssimulation	Börsenverhalten
Ausbildung	Angestelltenschulung	Schulung	Angestelltenschulung

Tabelle 2.1: Einsatzgebiete und Aufgabenbereiche von Expertensystemen (Nach [Neb 88]).

2.2 Einsatzbereich und Aufgabenbeschreibung von ZEX

Aufgabe der Bedienermannschaft des ZEUS-Experiments ist es, den Betrieb des Detektors zu gewährleisten und eine hohe Verfügbarkeit des Systems bereitzustellen. Die Operateure müssen die während der Datennahme auftretenden Probleme und Fehler schnellstmöglich erfassen, die Fehlerquelle lokalisieren und entsprechende Maßnahmen zur Fehlerbeseitigung einleiten. Die Bediener werden dabei durch eine Reihe von Hilfsmitteln unterstützt.

Zur Steuerung des Experiments, Überwachung des Systemstatus und Festlegung der System- und Triggerkonfiguration steht Run Control (vergleiche Abschnitt 1.3) zur Verfügung. Slow Control (siehe Abschnitt 1.3) dient der Überwachung 'langsam' veränderlicher Detektorparameter wie Hochspannungsversorgungen, Temperaturen, Gasdrücken, Magnetfeldstärken etc. Zur Beurteilung und Prüfung der Datenqualität sind Online-Eventdisplays und Online-Histogramme wichtiger Systemparameter installiert. Desweiteren stehen die Überwachungsdaten einzelner Komponenten zur Verfügung.

Aufgabe der Operateure ist es, aus den zur Verfügung stehenden Informationen die relevanten zu selektieren und eine Interpretation der Situation vorzunehmen. Aufgrund der Vielzahl dargebotener Daten (die Informationsmenge verteilt sich über etwa zwanzig Bildschirme) und der begrenzten Aufnahmefähigkeit der Bediener, wird nur ein Bruchteil der möglichen Informationen wahrgenommen, so daß eine globale Interpretation des Systemzustandes oftmals unmöglich ist. Die zur Verfügung stehenden Informationen sind teilweise sehr speziell und erfordern ein dementsprechend umfangreiches und detailliertes Wissen. Dies trifft insbesondere auf die Überwachungsdaten der einzelnen Komponenten zu, so daß es bei unzureichender Kenntnis des jeweiligen Systems zu einer Mißinterpretation kommen kann. Für eine verlässliche Aussage ist in diesen Fällen die Konsultation der jeweiligen Komponentexperten erforderlich, was zu entsprechenden Zeitverlusten und damit verbundenen Ausfallzeiten des Detektors führt. Erschwerend kommt hinzu, daß die Systemexperten und deren Wissen nicht jederzeit verfügbar sind, wobei insbesondere die geringe Verweildauer der einzelnen Kollaborationsmitglieder ein Problem darstellt. Das Expertenwissen steht daher oftmals nur für einen begrenzten Zeitraum zur Verfügung und muß anschließend durch Ausbildung eines neuen Experten teuer wiederer-

worben werden.

Zur Überwindung dieser Probleme und zur Unterstützung der Operateure bietet sich der Einsatz eines Expertensystems an. Dessen Aufgabe wird es sein, die zur Verfügung stehenden Informationen zu analysieren und die Operateure mit bereits aufbereiteten Meldungen zu versorgen. Da ein Expertensystem im Gegensatz zu Menschen jederzeit in der Lage ist, die gesamte dargebotene Informationsmenge aufzunehmen und auszuwerten, ist es so einem System möglich, wesentlich exaktere Analysen vorzunehmen. Ein Expertensystem arbeitet im Vergleich zu menschlichen Experten vektoruell und nicht wie diese skalar. Das bedeutet, daß sich ein derartiges System nicht durch einige - eventuell unwichtige - vom Sollwert abweichende Meßgrößen beeinflussen läßt, sondern stets den vollständigen Satz an Informationen und somit das gesamte System betrachtet. Desweiteren läßt sich das Wissen der einzelnen Systemexperten in der Wissensbasis des Expertensystems verankern, so daß es jederzeit abrufbereit vorliegt und nie neu erworben werden muß. Weitere Vorteile eines Expertensystems liegen in der ständigen Präsenz, hohen Geschwindigkeit und einer konstanten Aufmerksamkeit (keine Konzentrationsmängel), wodurch entsprechend geringere Ausfallzeiten des Detektors erreichbar sind.

Aus dem bisher Dargelegten läßt sich der Einsatz- und Aufgabenbereich von ZEX beschreiben und eingrenzen. ZEX wird zur Überwachung des Datennahmesystems eingesetzt werden und bei auftretenden Fehlern anhand der verfügbaren Informationen eine Fehlerdiagnose und -beschreibung zur Verfügung stellen. Anhand dieser Meldungen kann die Bedienermannschaft oder ein in das Expertensystem integriertes Kontrollsystem Steuersequenzen zur Experimentkontrolle und Fehlerbehebung einleiten. All dies muß entsprechend den menschlichen Experten in Echtzeit realisiert werden, so daß entsprechende Anforderungen an die Geschwindigkeit des Systems zu stellen sind.

2.3 Problemanalyse

In dem komplexen Datenerfassungssystem des ZEUS-Detektors können vor und während der Datennahme eine Reihe von Fehlern auftreten. Diese Probleme lassen sich grob in drei Gruppen einteilen, wobei die typischen Ursachen und die Möglichkeiten der Fehlererkennung im folgenden kurz exemplarisch vorgestellt werden.

- Ein Starten des Datenerfassungssystems kann unmöglich sein. Typische Ursachen liegen in Komponentenausfällen, Protokollverletzungen oder Synchronisationsproblemen, so daß es vorkommt, daß eine Komponente Systemmeldungen in der falschen Reihenfolge oder gar nicht bekommt bzw. selbst falsche oder fehlerhafte Datenpakete oder -sequenzen sendet. Die Ursachen derartiger Fehler können in fehlerhafter Software, aber auch defekter Hardware oder Spannungsausfällen liegen. Um derartige Fehler erkennen zu können, ist es notwendig, die zwischen den Komponenten ausgetauschten Meldungen zu überwachen, sowie die Slow Control Informationen auszuwerten.
- Während eines Runs kann eine Unterbrechung oder gar ein Abbruch der Datennahme auftreten. Typische Ursachen sind der Ausfall eines Komponentensubsystems, die Blockade des Datentransfers vom TLT (Third Level Trigger, vergleiche Abbildung 1.2 auf Seite 7) zum Massenspeicher oder der Verlust einiger Datenpakete. Diese Probleme resultieren in einem Überlauf der Datenpuffer und führen zu einem Stillstand der Datennahme. Sie lassen sich durch umfangreiche Überwachungen, beispielsweise der Pufferauslastungen und des Datendurchsatzes, und der zwischen den Komponenten ausgetauschten Meldungen erkennen.

- Die Qualität der Daten kann für physikalische Analysen unzureichend sein. Die Ursachen können vielfältiger Natur sein, hier seien nur die Möglichkeiten einer Miskalibration, toter Datenkanäle oder fehlender Hochspannungen genannt. Diese Probleme ließen sich durch eine Überwachung der Online-Histogramme oder der Online-Event-Displays erkennen und die eventuellen Ursachen aus den Slow Control Informationen ableiten.

Entsprechend dem im Abschnitt 2.2 skizzierten Einsatzbereich lassen sich die Aufgaben des Expertensystems in die drei Bereiche aufteilen. Den ersten Aufgabenbereich erfüllt ein Überwachungssystem, welches die zur Verfügung stehenden Informationen in Echtzeit analysiert und eine Detektion und Identifikation kritischer Systemzustände vornimmt. Auf diesem System aufbauend läßt sich ein Diagnosesystem einsetzen, welches die vom Überwachungssystem gelieferten Symptome aufnimmt und anhand der zur Verfügung stehenden Überwachungsdaten eine Fehleranalyse und -diagnose vornimmt. Den dritten Bereich des Expertensystems bildet ein Kontrollsystem. Dieses hat die Aufgabe, den im Datennahmesystem aufgetretenen Fehler zu beheben und einen reibungslosen, durch geringe Ausfallzeiten gekennzeichneten Experimentierbetrieb zu gewährleisten. Das generelle Aussehen eines solchen, für den Einsatz beim ZEUS-Experiment geplanten Expertensystems, ist Abbildung 2.1 zu entnehmen.

Auffallend ist die weitestgehende Unabhängigkeit der Diagnose- und Kontrollmodule. Dies ist möglich und notwendig, da es sich im bisherigen Betrieb des ZEUS-Detektors gezeigt hat, daß in vielen Fällen eine Steuerung des Experiments auch erfolgen kann und muß, ohne daß eine Diagnose vorliegt. Zu diesem Zweck ist dem Kontrollsystem, wie in Abbildung 2.1 gezeigt, ein heuristisches Wissen zur Fehlerbeseitigung zur Verfügung zu stellen. Andererseits kann das Kontrollsystem nicht völlig losgelöst vom Diagnosesystem betrieben werden, da beim Vorliegen einer Diagnose eine entsprechende Reaktion eingeleitet werden muß.

Die Aufspaltung in drei unabhängige Module bietet insbesondere bei der Entwicklung des Expertensystems eine Reihe von Vorteilen. Die Module können nacheinander und unabhängig voneinander entwickelt werden. Es bietet sich an, mit einem Überwachungssystem zu beginnen, dieses weitestgehend zu optimieren, und in den folgenden Schritten das Diagnose- und ein Kontrollsystem zu entwickeln. Die Module lassen sich insbesondere dem Entwicklungsstand des zu kontrollierenden Systems anpassen, wodurch eine entsprechende Schwerpunktsetzung (mehr Diagnose in der Entwicklung, mehr Kontrolle im Betrieb) möglich ist. Die Weiterentwicklung der Systeme läßt sich unabhängig voneinander vornehmen, und der modulare Aufbau ermöglicht den Einsatz einer für das jeweilige System optimalen Programmiersprache. Die einzelnen Module werden in den folgenden Abschnitten näher vorgestellt, wobei eine Übersicht in Tabelle 2.2 zu finden ist.

2.3.1 Das Überwachungssystem

Die Aufgabe dieses Moduls besteht darin, den Strom von Überwachungsdaten zu analysieren und ein anomales Verhalten des Datenerfassungssystems zu detektieren. Zu diesem Zweck ist es notwendig, dem Modul umfangreiche und detaillierte Statusinformationen zur Verfügung zu stellen, wofür beispielsweise die Überwachungsdaten der einzelnen Komponenten, die Meldungen des Runsequencing, die Runkonfiguration, der DAQ-System-Status, die Online-Histogramme sowie die Informationen von Slow Control zur Verfügung stehen. Da sich Fehler in dem Datennahmesystem durch charakteristische Wertekombinationen der Überwachungsgrößen zeigen und somit bestimmte Muster im Datenstrom darstellen, bieten sich zum Erkennen anomaler Systemzustände die Methoden der Mustererkennung an. Dazu wird dem System zunächst

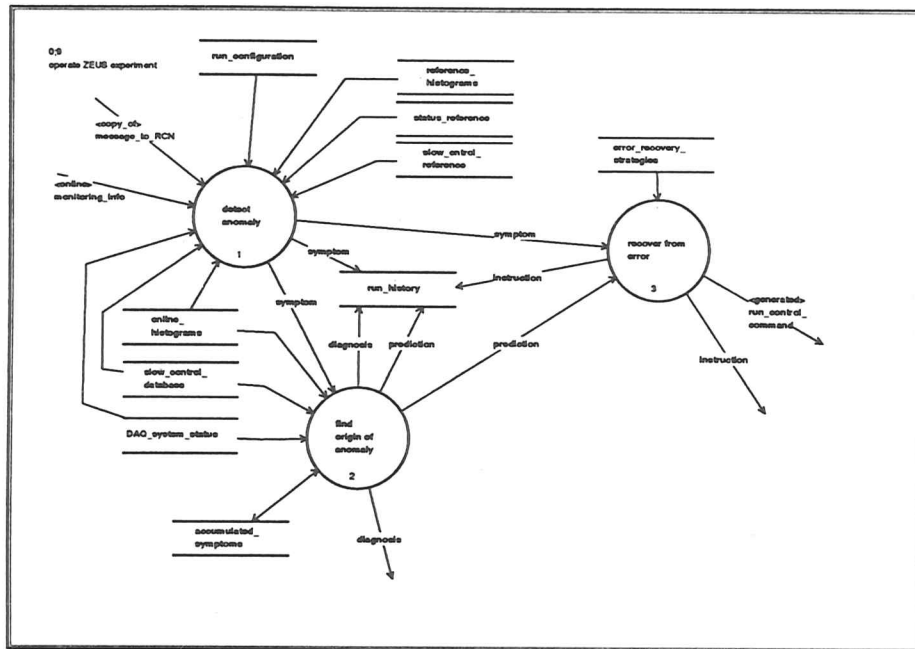


Abbildung 2.1: Datenflußdiagramm, welches die Architektur des ZEUS-Expertensystems zeigt. Die Kreise repräsentieren Prozesse, die Balken Datenspeicher und die Pfeile Datenflüsse. Zu erkennen ist die Aufspaltung in drei unabhängige Module, ein Überwachungs- (detect anomaly), ein Diagnose- (find origin of anomaly) und ein Kontrollsystem (recover from error). Nähere Erläuterungen zu den Prozessen finden sich in den Abschnitten 2.3.1 bis 2.3.3

	Überwachungssystem	Diagnosesystem	Kontrollsystem
Zielsetzung	Fehlererkennung	Fehlerquellen finden, Leistungsoptimierung	Fehlerbeseitigung
Methode	Mustererkennung	Backward Chaining Mustererkennung	Forward Chaining
Voraussetzung	umfangreiche Statusinformationen	intensive Überwachung	vielfältige Kontrollmöglichkeiten

Tabelle 2.2: Die drei Aufgabenbereiche und Module des Expertensystems, ihre Zielsetzungen, angewandten Methoden und Voraussetzungen. Eine genaue Beschreibung der Module und der Methoden ist in den Abschnitten 2.3.1 bis 2.3.3 zu finden.

ein Satz charakteristischer, typische Fehler repräsentierender Muster vorgegeben (in der Abbildung 2.1 durch die Slow Control-, Histogramm- und Statusreferenzen dargestellt). Diese werden mit den aktuellen Überwachungsdaten verglichen und mit Hilfe von, an dieser Stelle nicht näher

beschriebenen Verfahren³ wird versucht, die bekannten Muster wiederzuerkennen und eine Interpretation des Systemzustandes vorzunehmen. Bei dem Erkennen eines Fehlers wird eine entsprechende Meldung über den momentanen Systemzustand der Bedienermannschaft mitgeteilt, sowie das zugehörige Symptom sowohl an das Diagnose- als auch an das Kontrollsystem weitergeleitet.

2.3.2 Das Diagnosesystem

Dieses System dient dazu, aus den vom Überwachungssystem gelieferten Symptomen und den zur Verfügung stehenden Überwachungsdaten Fehlerdiagnosen abzuleiten und so zur Fehlerbeseitigung oder Systemoptimierung beizutragen. Die vom System gelieferten Diagnosen und Vorhersagen über das Systemverhalten werden der ZEUS-Bedienermannschaft und dem Kontrollsystem zugeführt. Um genaue Diagnosen durchführen zu können, ist die Bereitstellung möglichst vielfältiger und umfangreicher Überwachungsdaten erforderlich. Zur Implementation eines Diagnosesystems bieten sich unterschiedliche Methoden an. Zum einen ist es möglich, die Diagnose ebenfalls unter Zuhilfenahme der Methoden der Mustererkennung vorzunehmen. Eine andere Methode liegt im Aufbau einer Wissensbasis und der Anwendung eines bestimmten Schlußfolgerungsmechanismus, des sogenannten Backward Chaining. Dieser Mechanismus prüft anhand vorliegender Fakten mögliche Hypothesen (in diesem Fall die Diagnosen) auf ihren Wahrheitsgehalt. Dazu werden die zur jeweiligen Hypothese führenden und in der Wissensbasis abgelegten Regeln auf die Erfüllung aller notwendiger Fakten geprüft. Eine typische Programmiersprache für ein derartiges Vorgehen ist PROLOG.

2.3.3 Das Kontrollsystem

Ziel dieses Systems ist es, aus den Symptomen des Überwachungssystems und den eventuell vorliegenden Informationen des Diagnosesystems Kontrollsequenzen zur Steuerung des Experiments abzuleiten und der Experimentsteuerung zuzuführen oder Bedienungshinweise an die Operateure zu geben. Zur wirkungsvollen Arbeit des Systems sind vielfältige und umfangreiche Kontrollschnittstellen erforderlich, da nur so eine präzise, auf die jeweilige Situation abgestimmte Steuerung des DAQ-Systems möglich ist. Die Implementation eines Kontrollsystems ist über eine Wissensbasis möglich, wobei die zur Anwendung kommende Schlußfolgerungsmethode das Forward Chaining ist. Dabei handelt es sich um das dem Backward Chaining entgegengesetzte Vorgehen, d. h. es werden alle bekannten Fakten gegen die existierenden Regeln geprüft und die mit den erfüllten Regeln verbundenen Aktionen ausgeführt. Typisches Beispiel einer Programmiersprache für Kontrollsystem ist OPS 5.

2.4 Ein Entwurf für das Expertensystem ZEX

In den dem Expertensystem zur Verfügung stehenden Überwachungsdaten wird nach Fehlersymptomen gesucht, aus welchen dann eine Diagnose oder Steueranweisung gewonnen werden kann. Während die Berechnung und Analyse von Überwachungsdaten determiniert ist, trifft dies auf die Gewinnung von Diagnosen oder Kontrollsequenzen nicht notwendigerweise zu. Aus

³Die im Prototyp eingesetzten Verfahren werden in Abschnitt 3.3.2 ausführlich vorgestellt. Eine mathematisch exaktere Darstellung der verwendeten Verfahren findet sich in [Beh92b]

diesem Grund kann das Expertensystem in drei unterschiedliche Verarbeitungsstufen aufgeteilt werden: Voranalyse (Preprocessing), allgemeine Analyse (General Purpose Analysis) und fallspezifische Analyse (Case Specific Analysis).

Die Voranalyse nimmt eine Umwandlung und Selektion der verfügbaren Informationen vor, wodurch den Zustand des Datenerfassungssystems möglichst eindeutig beschreibende Informationen gewonnen werden. Bei diesen Daten kann es sich sowohl um einzelne Werte als auch um Wertekombinationen der Überwachungsdaten handeln.

Die allgemeine Analyse operiert auf den aus der Voranalyse gewonnenen Daten und prüft diese auf einen vorhandenen, anomalen Systemzustand. Dabei werden die Methoden der Mustererkennung eingesetzt, deren Algorithmen sehr universell sind und auf alle Überwachungsdaten angewandt werden können.

Die fallspezifische Analyse sucht unter Berücksichtigung der Experimentbedingungen und des aktuellen Systemzustandes nach der Ursache der aufgetretenen Symptome und leitet eine Diagnose, eventuell eine Vorhersage über zukünftiges Systemverhalten, sowie Hinweise zur Bedienung des Experiments ab.

Die Vor- und allgemeine Analyse entspricht dem in Abschnitt 2.3.1 vorgestellten Überwachungsprozeß, während die fallspezifische Analyse in den Diagnose- und Kontrollsystemen der Abschnitte 2.3.2 und 2.3.3 stattfindet.

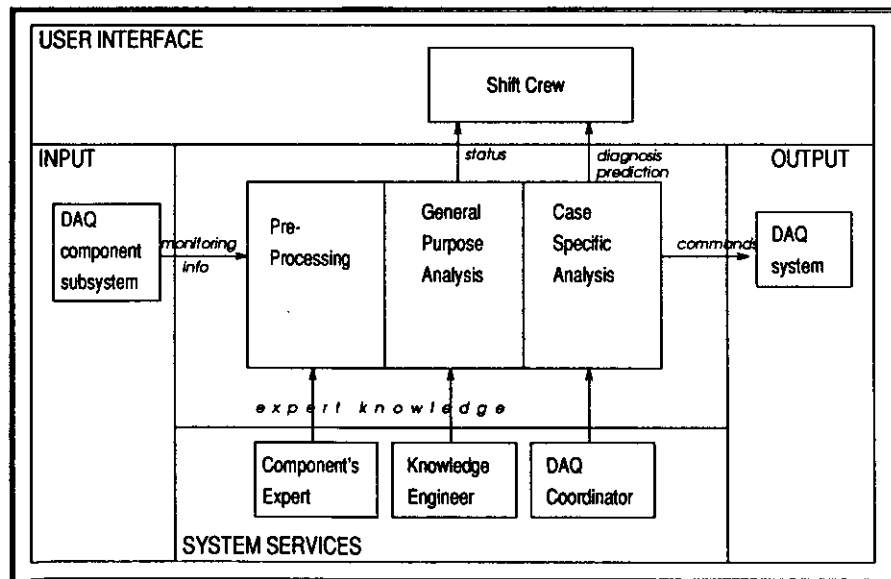


Abbildung 2.2: Die Systemschnittstellen des Expertensystems (Aus [Beh 93]).

Die Einbindung des Expertensystems in das Online-Datenerfassungssystem und die erforderlichen Systemschnittstellen sind der Abbildung 2.2 zu entnehmen. Die Eingaben des Systems, d. h. die Überwachungsdaten werden dem Expertensystem von den Komponentensubsystemen zur Verfügung gestellt. Die Ausgaben des Expertensystems bestehen einerseits in Kontrollkommandos zur Steuerung des DAQ-Systems, andererseits aus Status-, Diagnose- und Vor-

hersagemeldungen, die der Bedienermannschaft des Experiments über eine Benutzeroberfläche dargeboten werden. Zur Wartung, Erweiterung und Überprüfung des Systems werden Service-schnittstellen vorgesehen.

Zur Entwicklung und Optimierung des Expertensystems ist das Wissen einer Vielzahl von Experten erforderlich. Die Experten der Komponentensubsysteme verfügen über genaue Kenntnisse der Überwachungsdaten 'ihres' Systems und wissen, welche Werte oder Wertekombinationen zur Beschreibung des jeweiligen Systems relevant sind. Über das zur Analyse der Überwachungsdaten benötigte Wissen verfügt der Wissensingenieur (Knowledge Engineer). Der DAQ-Koordinator schließlich besitzt die Kenntnisse über das Zusammenwirken der Komponenten, Fehlerursachen, nötige Kontrollsequenzen und den Gesamtstatus des Systems.

Wie Abbildung 2.2 zeigt, wird das Wissen der einzelnen Experten in unterschiedlichen Teilen des Systems gespeichert. Während einerseits die Komponentexperten das für die Voranalyse benötigte Wissen besitzen, stellen die DAQ-Koordinatoren die Kenntnisse für die fallspezifische Analyse zur Verfügung. Das Wissen eines Wissensingenieurs ist nach der durch ihn erfolgten Implementation der allgemeinen Analyse nicht mehr erforderlich, so daß eine Wartung des Systems allein durch die Komponenten- und DAQ-Experten durchgeführt werden kann. Dies ist eine wichtige Eigenschaft, da derartige Experten über die gesamte Lebensdauer des Experiments verfügbar sind.

Kapitel 3

Entwurf und Implementation eines Prototypen: ZEX-P

Als Realisierbarkeitsstudie des Expertensystems wurde zunächst ein Prototyp entwickelt und in Betrieb genommen. Ziel dabei war es, die Möglichkeiten und Grenzen der zugrunde liegenden Verfahren zu ermitteln und Erfahrungen für die Implementation des endgültigen Systems zu sammeln.

Die Entwicklung des Prototypen erfolgte auf Basis der vom Eventbuilder bereitgestellten Überwachungsdaten sowie den Steuersequenzen der zentralen Experimentsteuerung. Bei der Entwicklung des Prototypen wurde ein Überwachungs- sowie Diagnosemodul implementiert und auf die Entwicklung eines umfangreichen Kontrollsystems verzichtet. Das Kontrollsystem beschränkt sich auf die Bereitstellung von einfachen Hinweisen an die Operateure und führt keine eigenständige Steuerung des Experiments durch.

3.1 Die Arbeitsumgebung

Als Arbeitsumgebung für den Prototypen stand eine Silicon Graphics Workstation¹ zur Verfügung, die auch als Entwicklungsumgebung für den Eventbuilder dient. Die Workstation ist über einen Transputerlink² mit dem Eventbuilder und ein LAN (Local Area Network) mit der zentralen Experimentsteuerung verbunden, so daß auf diesem Rechner die Überwachungsdaten des Eventbuilders sowie die Informationen der zentralen Systemsteuerung zur Verfügung stehen.

Die Entwicklung des Expertensystemprototypen wurde auf Basis der Überwachungsdaten des Eventbuilders durchgeführt, da es sich bei dem Eventbuilder um ein sehr gut untersuchtes System handelt, welches eine Vielzahl von Überwachungsdaten zur Verfügung stellt [Sch 92] und dessen zentrale Position innerhalb des Datenerfassungssystems (vergleiche Abbildung 1.2) die Bestimmung aussagekräftiger Systemparameter erlaubt. Dies trifft um so mehr zu, als Datenverbindungen zu allen Detektorkomponenten bestehen und daher Informationen über den Zustand des gesamten Datennahmesystems verfügbar sind. Desweiteren hat sich im Laufe des bisherigen Experimentierbetriebes gezeigt, daß der Eventbuilder eine sehr zuverlässig arbeitende Komponente darstellt, was nicht zuletzt auf die bei dessen Entwicklung eingesetzte Methode des SA/SD (Structured Analysis, Structured Design) [You 89] zurückzuführen ist [Beh92c, Beh92d].

¹Silicon Graphics 4D/25 unter UNIX

²Mit 'Link' werden die auf dem Chip integrierten seriellen Schnittstellen einer speziellen Mikroprozessorfamilie, den Transputern bezeichnet.

Bevor näher auf die vom Eventbuilder zur Verfügung gestellten Überwachungsdaten eingegangen wird, erfolgt zunächst eine kurze Beschreibung des Aufbaus des Eventbuilders, wobei Details [Beh 92, Sch 92] zu entnehmen sind.

Entsprechend seiner Aufgabe, dem Sammeln der zusammengehörigen Ereignisteildaten und dem Weiterleiten der daraus gebildeten Events an den TLT, verfügt der EVB über siebzehn Schnittstellen zu den Komponenten und sechs Schnittstellen zum TLT (vergleiche Abschnitt 1.2). Die Verbindung zwischen den Komponenten- und TLT-Schnittstellen wird über einen Kreuzschienenverteiler hergestellt, der in der Lage ist, jede Komponentenschnittstelle mit jeder TLT-Schnittstelle zu verbinden und in Eigenarbeit entwickelt wurde. Neben der Verbindung zu den Komponenten und dem TLT ist eine Anbindung des Eventbuilders an den GSLT (Global Second Level Trigger, vergleiche Abschnitt 1.2) vorhanden. Dies ist notwendig, da der GSLT nicht über Verbindungen zu allen Komponenten verfügt und die von ihm getroffenen Entscheidungen über den Eventbuilder den Komponentensubsystemen zugeleitet werden. Der gesamte Datenfluß innerhalb des Eventbuilders, das Verteilen der GSLT Entscheidung an die Komponenten, das Schalten des Kreuzschienenverteilers und das Transferieren der Komponentendaten zu den TLT Schnittstellen wird von einer Steuereinheit überwacht. Einen Überblick über die Funktionseinheiten des Eventbuilders und deren Zusammenwirken vermittelt die Abbildung 3.1.

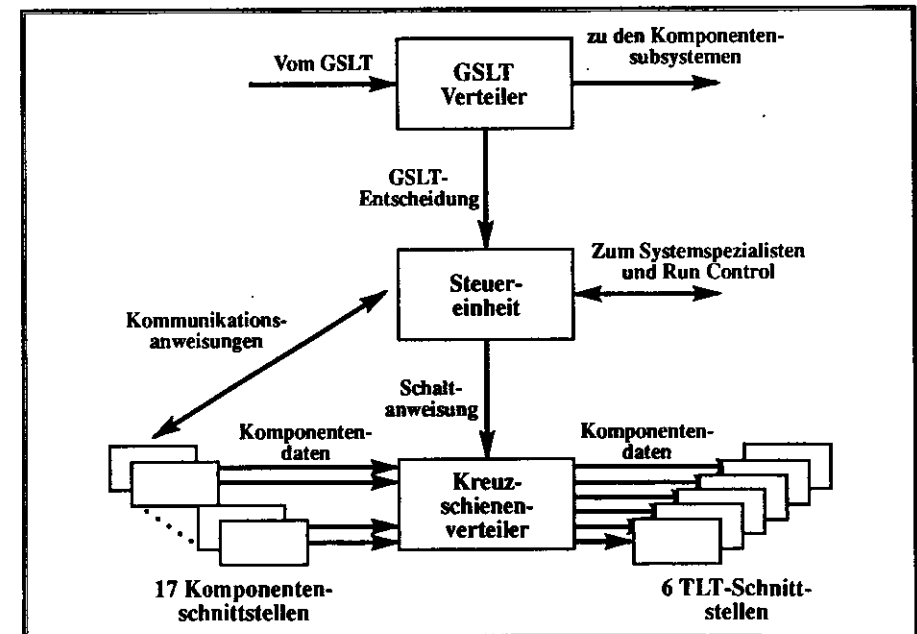


Abbildung 3.1: Die Funktionseinheiten des Eventbuilders (Nach [Sch 92]).

Der Eventbuilder wurde als Transputernetzwerk realisiert und basiert auf den von der Kollaboration entwickelten 2TP-VME-Modulen. Jedes dieser Module enthält zwei Transputer vom

Typ T800, eine VMEbus-Schnittstelle³ sowie einen allen drei Bestandteilen gemeinsamen Speicher, auf den simultan zugegriffen werden kann, das sogenannte Triple Ported Memory (TPM). Dieser Speicher wird in Form eines Ringpuffers verwaltet und verfügt über eine Kapazität von 512 kBytes. Auf die Funktionsweise der Komponenten und TLT-Schnittstellen wird nachfolgend näher eingegangen.

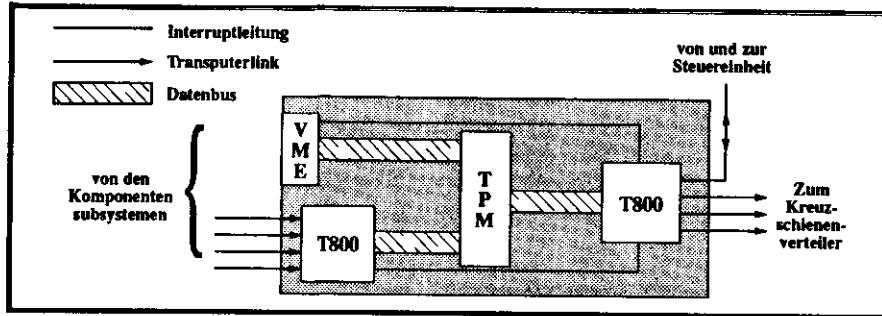


Abbildung 3.2: Das 2TP-VME-Modul als Komponentenschnittstelle (Aus [Sch 92]).

Wie der Abbildung 3.2 zu entnehmen ist, wird beim Einsatz eines 2TP-VME-Moduls als Komponentenschnittstelle lediglich einer der beiden Transputer vom Eventbuilder genutzt, so daß dem Komponentensubsystem ein Transputer zur freien Verfügung steht. Durch die Verfügbarkeit sowohl des Transputers als auch der VME-Schnittstelle seitens des Komponentensubsystems ist es sowohl den auf VME68k, als auch den auf Transputern basierenden Komponenten möglich, ihre Daten im TPM abzulegen und so an den Eventbuilder zu übergeben. Das Vorliegen der Daten im TPM wird dem vom Eventbuilder genutzten Transputer durch eine Interruptleitung angezeigt, woraufhin dieser die Daten aus dem TPM in seinen internen, 3 MBytes umfassenden Datenpuffer einliest. Sobald die Steuereinheit die zu diesen Daten gehörige Triggerentscheidung erhalten hat, wird über den Kreuzschienenverteiler eine Verbindung zu einem freien TLT-Knoten hergestellt. Nach erfolgreichem Transfer der Daten wird die Verbindung wieder gelöst und der Zyklus kann von neuem beginnen.

Der Einsatz eines 2TP-VME-Moduls als TLT-Schnittstelle ist Abbildung 3.3 zu entnehmen. Hier stehen beide Transputer dem Eventbuilder zur Verfügung und die Verbindung zum TLT wird über die VME-Schnittstelle hergestellt. Die von den Komponenten über den Kreuzschienenverteiler gesandten Ereignisteildaten werden von einem der unabhängig voneinander operierenden Transputern eingelesen, wobei die zu einem Event gehörigen Teildaten auf den jeweils gleichen Transputer transferiert werden. Die empfangenen Daten werden zunächst in den internen, 3 MBytes umfassenden Puffer geschrieben und dort bis zum Vorliegen aller Ereignisteildaten gespeichert. Nachdem die Steuereinheit die Vollständigkeit des Ereignisses signalisiert hat, werden die empfangenen Daten in eine ADAMO-Datenstruktur⁴ umgewandelt, in das TPM kopiert und dort bis zur Abnahme durch den TLT gespeichert.

Während des Betriebes des Eventbuilders werden eine Vielzahl von Parametern, wie die Frequenz positiver GSLT-Entscheidungen, die Abnahmezeit der Komponenten für Triggerent-

³VMEbus ist eine von einer Gruppe von Rechnerherstellern 1981 entworfene und inzwischen zu einem weit verbreiteten Industriestandard gewordene Rechnerarchitektur [Pet 89].

⁴Die ZEUS-Kollaboration entschied sich, Daten in dem ADAMO Format abzulegen. Eine Beschreibung dieses Formats findet sich in [ADA 91].

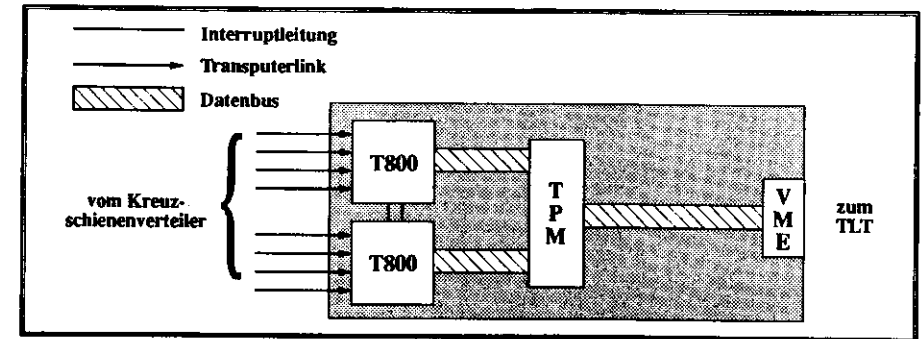


Abbildung 3.3: Das 2TP-VME-Modul als TLT-Schnittstelle (Aus [Sch 92]).

scheidungen, die Länge der Ereignisteildaten, der Datendurchsatz an der Komponentenschnittstelle, oder die Füllzustände der Übergabepuffer der Komponentensubsysteme sowie der TLT-Schnittstellen bestimmt und auf Datenträgern gespeichert. Eine Auswahl der aussagekräftigsten Daten findet sich in Tabelle 3.1, wobei eine genaue Beschreibung der zur Verfügung stehenden Überwachungsdaten und der daraus gewonnenen Ergebnisse [Sch 92] zu entnehmen ist.

3.2 Die Systemanalyse

Vor dem Beginn der Implementation des Prototypen wurden zunächst intensive Analysen der Ende des Jahres 1992 gewonnenen und auf Datenträgern verfügbaren EVB-Überwachungsdaten durchgeführt. Bei diesen Betrachtungen konnten typische Fehler in der Datennahme und Schwachpunkte des Datenerfassungssystems erkannt werden.

Eine häufige Fehlerursache war das während der Datennahme auftretende Fehlverhalten einer Detektor Komponente (Abbildung 3.4), welches sich zunächst durch ein Füllen der internen Datenpuffer der TLT-Schnittstelle (I_{TLT}) bemerkbar macht. Die Ursache liegt in einem Ausfall oder einer extrem langen Antwortzeit der Komponente auf die GSLT-Entscheidungen ($d_{K,n}$, im Bereich einiger Sekunden), so daß während dieses Zeitraums unvollständige Ergebnisse auf der TLT-Schnittstelle gespeichert werden müssen. Ist die Zeitdifferenz zwischen den GSLT-Entscheidungen ($\Delta t_{T,i,g,n}$) klein gegenüber den Zeitabständen zwischen dem Transfer der Komponentendaten ($\Delta d_{K,n}$), so führt dies innerhalb kürzester Zeit zu einer hundertprozentigen Pufferauslastung der TLT-Schnittstelle. Da von diesem Zeitpunkt an keine Daten mehr von den Komponentensubsystemen zum TLT transferiert werden können, beginnen die Puffer der Komponentenschnittstellen (I_K) vollzulaufen, was schließlich in einem Stillstand der Datennahme resultiert. Das angesprochene Problem trat vorwiegend bei den Komponenten auf, die eine große Datenmenge zu bewältigen haben, den Kalorimetern (BCAL, FCAL, RCAL) und der zentralen Spurkammer (CTD). Die Analysen zeigten, daß dieses Problem selbst bei dem Ausfall nur einer Komponente und niedriger GSLT-Frequenz ($\approx 30 \text{ Hz}$) innerhalb weniger Sekunden zu einem Stillstand des Datennahmesystems führt. Oftmals traten die Probleme bei mehreren der kritischen Komponenten (BCAL, FCAL, RCAL, CTD) gleichzeitig auf, so daß sich das Füllen der Datenpuffer über einen längeren Zeitraum erstreckt und der Stillstand entsprechend später eintritt. Dieses zunächst widersprüchliche Verhalten läßt sich folgendermaßen

Größe	Messung bzw. Berechnung	Bedeutung
$t_{Trig,n}$	Meßgröße	Zeitpunkt des Eintreffens der positiven Triggerentscheidung für Ereignis n.
f_{Trig}	$ \{(n t_{Trig} \in [t_0, t_1])\} $ $t_1 - t_0$	GSLT-Frequenz im Intervall $[t_0, t_1]$.
$t_{K,n}$	Meßgröße	Zeitpunkt des Eintreffens der Ereignisteildaten von Komponente K für Ereignis n.
$\Delta t_{K,n}$	$t_{K,n} - t_{K,n-1}$	Abstand zweier aufeinanderfolgender Ereignisteildaten.
$d_{K,n}$	$t_{K,n} - t_{K,n-1}$	Antwortzeit der Komponente K auf eine GSLT-Entscheidung.
$I_{K,n}$	Meßgröße	Prozentualer Füllzustand des internen Pufferspeichers der Schnittstelle von Komponente K zum Zeitpunkt $t_{K,n}$.
$u_{TLT}(n)$	Meßgröße	Nummer des TLT-Schnittstellentransputers auf dem das Ereignis n konstruiert wurde.
$I_{TLT,u,n}$	Meßgröße	Prozentualer Füllzustand des internen Pufferspeichers des TLT-Schnittstellentransputers TLTu nach Empfang der letzten Ereignisteildaten zu Ereignis n.
$P_{TLT,u,n}$	Meßgröße	Prozentualer Füllzustand des TPM der TLT-Schnittstelle TLTu nach Empfang der letzten Ereignisteildaten zu Ereignis n.

Tabelle 3.1: Die Offline-Überwachungsdaten (Eine Auswahl aus [Sch 92]).

verstehen: Der Stillstand tritt ein, sobald eine Komponente den ihr verfügbaren Speicherbereich vollständig gefüllt hat. Der Zeitraum bis zum Stillstand wird folglich durch den Quotienten des verfügbaren Speicherbereichs und der höchsten Datenrate einer funktionierenden Komponente bestimmt. Bei einem Ausfall der kritischen, die größte Datenmenge transferierenden Komponenten wird der Zeitraum daher entsprechend größer.

Ein weiteres, wesentlich häufiger auftretendes Problem stellt die Blockade oder geringe Transfargeschwindigkeit des TLT dar (Abbildung 3.5). Die TLT-Schnittstelle stellt den kritischsten Punkt im Datenerfassungssystem dar, da auf Seiten des Massenspeichers die Transfargeschwindigkeit auf 0,5 MBytes pro Sekunde begrenzt ist und der Detektor im Normalbetrieb bereits an diesem Limit betrieben wird. Bei kurzzeitig, aufgrund von Fluktuationen in der GSLT-Frequenz auftretenden höheren Datenraten, kommt es zu einem sofortigen Datenstau. Die TLT-Probleme lassen sich an einem Füllen der TPM-Puffer der TLT-Schnittstelle (P_{TLT}) erkennen, was zu einem nahezu sofortigen Überlauf der internen Datenpuffer der TLT-Schnittstelle (I_{TLT}) führt. Das Füllen des TPMs ist eindeutiges Indiz für Probleme bei der Datenabnahme durch den TLT, da Komponentenprobleme stets zu unvollständigen Ereignissen führen und die Daten daher nie in das TPM gelangen können. Im Gegensatz zu dem zuvor geschilderten Problem verringert sich hier mit zunehmender Zahl der ausfallenden TLT-Schnittstellen die Zeit bis zum Systemstillstand nicht, so daß eine sehr schnelle Reaktion erforderlich ist.

Die bisher geschilderten Probleme sind sehr sensitiv gegenüber der GSLT-Frequenz, d. h.

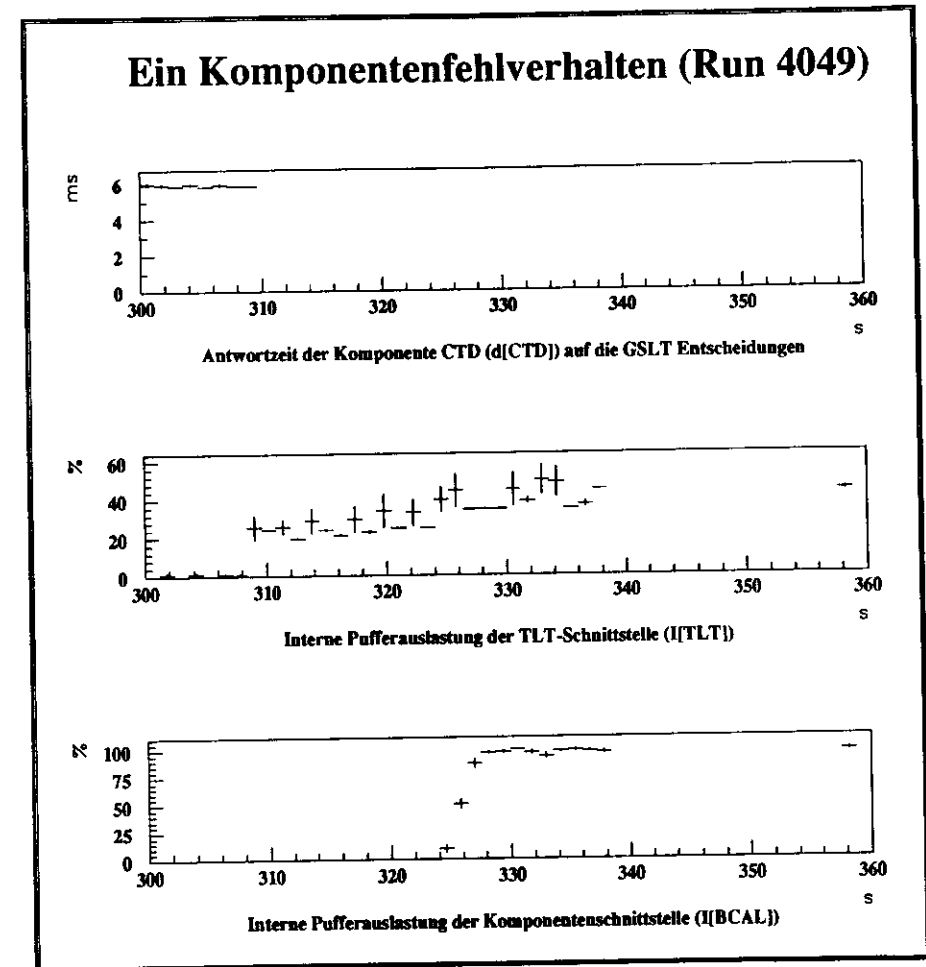


Abbildung 3.4: Ein Komponentenfehlerverhalten während des Runs 4049. Gegenüber der Zeit sind die Komponentenantwortzeit auf die GSLT-Entscheidungen (d_{CTD}), die interne Pufferauslastung der TLT-Schnittstelle (I_{TLT}) sowie einer ausgewählten Komponente (I_{BCAL}) aufgetragen. Etwa 310 Sekunden nach dem Beginn der Datennahme fällt die Komponente CTD aus. Dies führt innerhalb von wenigen Sekunden zu einem Anstieg der Pufferlast der TLT-Schnittstelle, was dann zu einem Anstieg der Speicherauslastung auf der Komponentenseite und schließlich zum Stillstand der Datennahme (der Zeitraum von 338 bis 358 Sekunden) führt.

insbesondere bei hohen Frequenzen treten die geschilderten Probleme vermehrt auf. Um die angesprochenen Probleme erkennen zu können, ist die Überwachung der nachfolgend angeführten Parameter notwendig:

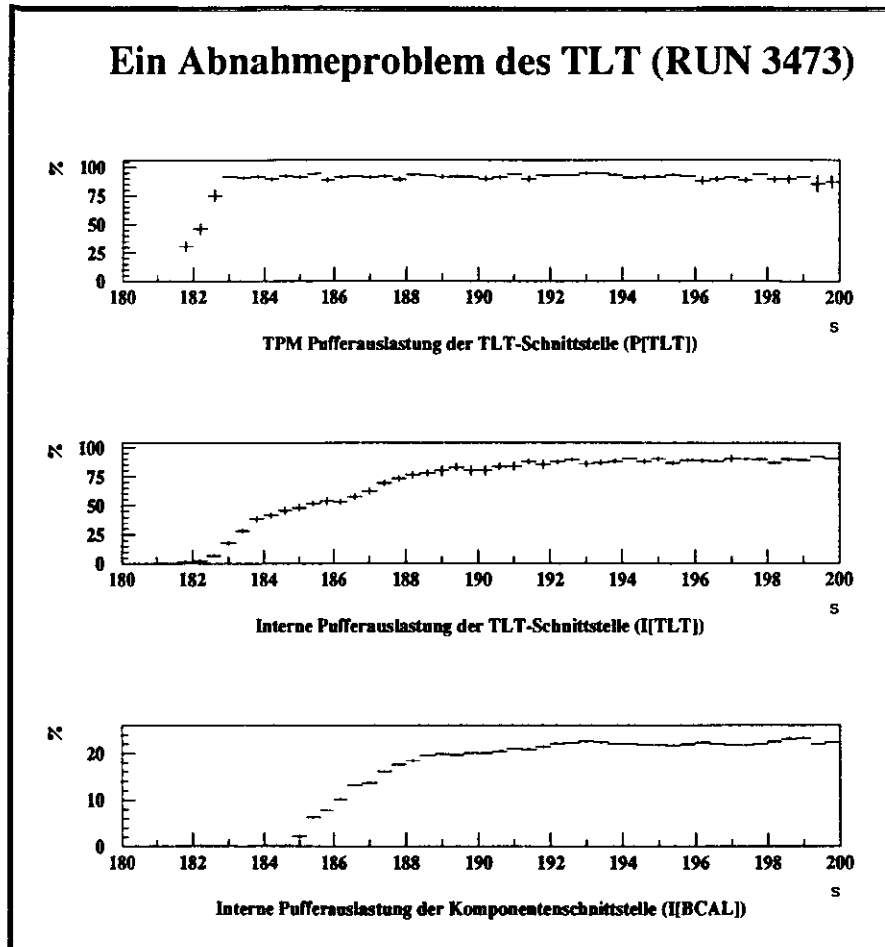


Abbildung 3.5: Ein Fehler in der Datenabnahme durch den TLT. Gegenüber der Zeit ist die Pufferauslastung des TPM (P_{TLT}) und des internen Datenspeichers der TLT-Schnittstelle (I_{TLT}) sowie exemplarisch des internen Datenpuffers einer Komponentenschnittstelle (I_{BCAL}) aufgetragen. 183 Sekunden nach dem Beginn der Datennahme kam es auf der TLT Seite zu einem Abnahmeproblem, was sich in einem Füllen der TPM-Puffer äußert. Innerhalb kürzester Zeit führt dies zu einem Rückstau der Daten und einem Füllen der internen TLT- sowie der Komponentenspeicher.

- Füllzustand der TLT Übergabepuffer ($P_{TLT_{u,n}}$, $I_{TLT_{u,n}}$)
- Füllzustand der Komponentenübergabepuffer ($I_{K,n}$)

- Abnahmezeit der Komponenten für Triggerentscheidungen ($d_{K,n}$)
- GSLT-Frequenz (f_{Trig})

Ziel des Prototypen ist es, die bekannten und typischen Fehler zuverlässig und schnell zu erkennen. Da hierfür einerseits umfangreiche Analysen der Überwachungsdaten vorzunehmen und andererseits in kritischen Situationen schnelle Reaktionen erforderlich sind, ist eine Aufteilung des Überwachungssystems in zwei getrennte Module erforderlich. Dabei wird ein System zur Überwachung der kritischen Punkte des Datennahmesystems (TLT-Schnittstelle) eingesetzt und operiert nur auf den Parametern $P_{TLT_{u,n}}$ und $I_{TLT_{u,n}}$. Aufgrund der geringeren Datenmenge und den daher schnellen Analysen ist dieses System in der Lage, innerhalb kürzester Zeit zu reagieren. Die weiterhin notwendigen Untersuchungen der gesamten Datenmenge werden dann von dem zweiten, entsprechend langsameren Modul ausgeführt. Diese Aufspaltung findet sich praktisch in jedem Echtzeit-Expertensystem, da nur so eine sowohl schnelle als auch präzise Reaktion möglich ist.

Die erforderliche Kommunikation des Expertensystems mit dem Bediener erfolgt über eine X-Windows-Oberfläche. Dem Benutzer werden dabei der Status des Datenerfassungssystems, die Fehlerdiagnose, die Vorhersagen, die zu ergreifenden Maßnahmen sowie die wichtigsten, das DAQ-Verhalten beschreibenden Parameter dargestellt. Es ist nicht möglich, seitens der Bediener Einfluß auf die Funktionsweise des Systems zu nehmen, so daß sich die Benutzeroberfläche auf eine reine Anzeige beschränkt.

3.3 Die Implementation von ZEX-P

Nachdem die Rahmenbedingungen bekannt sind, wird im nachfolgenden die konkrete Realisierung des Prototypen vorgestellt, wobei insbesondere die dabei zur Anwendung kommenden Methoden vorgestellt werden. Die Programmierung der Prozesse des Prototypen wurde vollständig in C durchgeführt.

3.3.1 Die Online-Überwachungsdaten

Die aufgezeichneten und vom Eventbuilder gelieferten Überwachungsdaten sind in der in [Sch 92] beschriebenen Form für Offline-Analysen vorgesehen, so daß für den Einsatz in einem Online-System entsprechende Änderungen notwendig wurden. Grund dafür ist die Tatsache, daß die Daten eventorientiert genommen wurden, d. h. die zu einem Event und den jeweiligen Ereignisteildaten gehörigen Werte wie Pufferauslastungen oder Komponentenantwortzeiten werden in einer gemeinsamen, dem Event zugeordneten Struktur erfaßt und stehen erst mit der Vollständigkeit des Ereignisses zur Verfügung. Da der EVB asynchron operiert, die Ereignisteildaten und zugehörigen Überwachungsdaten daher in beliebiger Reihenfolge anfallen, werden die zu einem Event gehörigen Informationen zunächst im Eventbuilder gesammelt und erst nach einer gewissen Anzahl von Events (zur Zeit 1024) komplett in einem Block herausgeschrieben. Dieses skizzierte Verfahren ist für ein Online-System nicht praktikabel, da es beim möglichen Fehlen von Eventdaten zum Verlust der jeweiligen Überwachungsdaten kommt und insbesondere bei Verzögerung einer Komponente oder einem Systemstillstand die Informationen auch erst mit der entsprechenden Verzögerung oder gar nicht zur Verfügung stehen. Es ist daher notwendig, die Datennahme nicht eventorientiert, sondern periodisch durchzuführen. Als Datennahmeintervall wurde ein Zeitraum von 196 ms gewählt, so daß mit einer Rate von etwa

5 Hz ein vollständiger Datensatz mit den neuesten Informationen zur Verfügung steht. Desweiteren wurde die Struktur der Überwachungsdaten vollständig überarbeitet. Bei den Online zur Verfügung stehenden Daten, die in Tabelle 3.2 zusammengefasst sind, handelt es sich um die folgenden Informationen:

- Als globale Information steht der Zeitpunkt der letzten während des betrachteten Datennahmeintervalls aufgetretenen positiven GSLT-Triggerentscheidung sowie deren Gesamtzahl während dieses Intervalls zur Verfügung. Weiterhin wird die Anzahl ununterbrochen aufeinander folgender, unmittelbar vorausgegangener Intervalle ohne jeglichen Datentransfer bestimmt.
- Als Information auf der Komponentenseite steht zunächst für jede Komponente die letzte, während des Datennahmeintervalls bekannte interne Pufferauslastung zur Verfügung. Desweiteren wird für jede Komponente die im Verlauf des Intervalls maximal auftretende Antwortzeit auf die GSLT-Triggerentscheidungen bestimmt. Bei diesem Wert sind zwei Fälle zu unterscheiden: Die GSLT-Entscheidung zur Komponentenantwort kann sowohl innerhalb als auch außerhalb des betrachteten Zeitraums liegen. Als weiterer Parameter steht die Gesamtzahl der Ereignisse zur Verfügung, für die sowohl GSLT-Entscheidung als auch Komponentenantwort im betrachteten Intervall lagen. Außerdem wird die Anzahl ununterbrochen aufeinander folgender, unmittelbar vorausgegangener Intervalle ohne jegliche Daten seitens des Komponentensubsystems bestimmt.
- Auf TLT-Seite stehen für jeden TLT-Transputer Informationen über die letzte während des Datennahmeintervalls bestimmte Pufferauslastung des internen, sowie des TPM-Speichers zur Verfügung. Auch hier existiert für jeden TLT-Transputer ein Zähler zur Bestimmung der Anzahl ununterbrochen aufeinander folgender, unmittelbar vorausgegangener Zeitintervalle ohne jegliche Daten für den jeweiligen Knoten.

3.3.2 Diskussion der verwendeten Methoden

Zur Identifikation des Systemzustandes durch das Überwachungssystem werden die in Abschnitt 2.3.1 angedeuteten Verfahren der Mustererkennung eingesetzt. Im Prototypen kommen drei unterschiedliche Methoden zur Anwendung, die sich an jeweils anderen Aufgabenbereichen orientieren.

Die einfachste Methode der Mustererkennung stellt die Überwachung von Schwellwerten dar, wo es sich bei den zu erkennenden Mustern um Über- bzw. Unterschreitungen dieser Grenzwerte handelt. Die Schwellwerte sind nicht notwendigerweise statisch, sondern können von der jeweiligen Situation abhängen, d. h. Funktionen mehrerer Parameter sein. Diese Methode ist sehr schnell und bildet oftmals den Auslöser für komplexe, weiterführende Analysen.

Neben der Überwachung einzelner Schwellwerte ist eine Untersuchung der gesamten vorliegenden Datenmenge erforderlich. Um daraus eine Interpretation des Systemzustandes vornehmen zu können, d. h. die bekannte Zustände repräsentierenden Muster im Datenstrom zu erkennen, wird die Methode der Cluster Zentren angewandt. Die Vorgehensweise dieses Verfahrens ist in Abbildung 3.6 illustriert.

Aufgabe des Systementwicklers ist es zunächst, aus den zur Verfügung stehenden Überwachungsdaten jene charakteristischen Parameter auszuwählen, welche zur eindeutigen Identifikation des Systemzustandes, dem zu erkennenden Muster, bestmöglich geeignet sind. So könnte

Größe	Bedeutung
$t_{Trig,n}$	Zeitpunkt der letzten GSLT Entscheidung während des Intervalls n
n_n	Anzahl der positiven GSLT Entscheidungen während des Intervalls n
d_{yc}	Anzahl aufeinanderfolgender Datennahmeintervalle ohne jeglichen Datentransfer
$I_{K,n}$	Letzter prozentualer Füllzustand des internen Pufferspeichers der Komponente K im Intervall n
$d_{K,n}^{max}$	maximale Antwortzeit der Komponente K auf die GSLT-Triggerentscheidung. Sowohl Triggerentscheidung als auch Komponentenantwort liegen im Intervall n
$d_{K,n}^{max,late}$	Maximale Antwortzeit der Komponente K im Intervall n auf Triggerentscheidungen aus einem Intervall $< n$
$t_{K,n}^{late}$	Zeitpunkt der zu $d_{K,n}^{max,late}$ gehörigen Triggerentscheidung
$C_{K,n}$	Anzahl der Ereignisteildaten der Komponente K, deren GSLT Entscheidung und Komponentenantwort im Intervall n lagen
$C_{K,n}^{del}$	Anzahl aufeinanderfolgender Überwachungsintervalle ohne jegliche Ereignisteildaten der Komponente K
$I_{TLT,u,n}$	Letzter, während des Intervalls n bestimmter prozentualer interner Pufferfüllzustand des TLT-Transputers u
$P_{TLT,u,n}$	Letzter, während des Intervalls n bestimmter prozentualer TPM-Pufferfüllzustand des TLT-Transputers u
$C_{TLT,u,n}^{del}$	Anzahl aufeinanderfolgender Überwachungsintervalle ohne jegliche Daten für TLT Knoten u

Tabelle 3.2: Die Online-Überwachungsdaten. Nähere Erläuterungen folgen im Text.

zur Identifikation des Status des Datenerfassungssystems beispielsweise die Auswahl der Überwachungsdaten Triggerrate (f_{Trig}) und Pufferauslastung ($\sum I_{TLT} + \sum I_K + \sum P_{TLT}$) eine geeignete Wahl darstellen. Die Auswahl der charakteristischen Parameter (Eigenschaften, Feature), die sowohl einzelne Daten (f_{Trig}) als auch Funktionen verfügbarer Daten ($\sum I_{TLT} + \sum I_K + \sum P_{TLT}$) sein können, wird als Feature Extraction bezeichnet. Die Feature Extraction stellt das größte Problem in der Mustererkennung dar, da die Auswahl der charakteristischen Eigenschaften größtenteils auf der Intuition des Systementwicklers beruht und praktisch keine Algorithmen zur Automatisierung existieren. Die gewählten Eigenschaften spannen einen mehrdimensionalen Raum, den sogenannten Feature Space oder Eigenschaftsraum auf, in welchem anschließend alle weiteren Operationen ausgeführt werden. Im betrachteten Beispiel spannen die Triggerfrequenz f_{Trig} und die Pufferauslastung $\sum I_{TLT} + \sum I_K + \sum P_{TLT}$ den in Abbildung 3.6b dargestellten, zweidimensionalen Raum auf. Dieser Feature Space sollte von möglichst geringer Dimension sein und eine einfache Metrik besitzen⁵. In dem so gewonnenen Raum wird anschließend die Positi-

⁵Die Metrik ist nicht notwendig euklidisch, da die ausgewählten Eigenschaften nicht unbedingt linear, sondern beispielsweise diskret sein können. Zudem können einige der gewählten Eigenschaften korreliert sein, so daß der Raum nicht orthogonal ist.

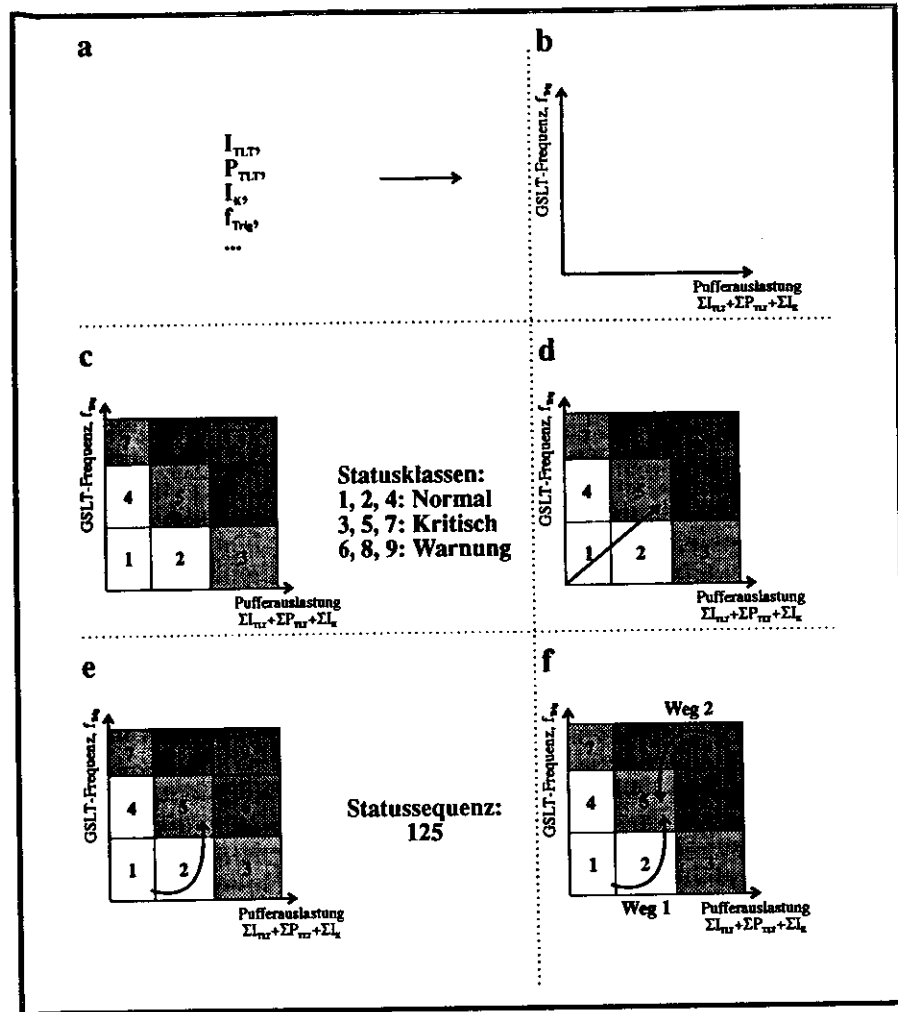


Abbildung 3.6: Die Methode der Cluster Zentren. Aus Überwachungsdaten (a) werden charakteristische Werte extrahiert und ein Feature Space gebildet (b). Systemzustände werden Clustern bzw. Regionen des Raums zugewiesen (c) und aus vorliegenden Überwachungsdaten wird ein Statusvektor (d) gebildet. Dieser Vektor wird dem nächstgelegenen Cluster zugeordnet, wobei ein kontinuierlicher Strom von Überwachungsdaten zu einem von den zugehörigen Statusvektoren beschriebenen Weg führt (e). Unterschiedliche Systemgeschichten führen zu abweichenden Wegen (f).

on der Wertekombinationen bekannter Systemzustände bestimmt. Bei einer geschickten Wahl

des Eigenschaftsraums bilden die zu gleichen Systemzuständen gehörigen Wertekombinationen Gruppen, sogenannte Cluster aus, denen dann, wie in Abbildung 3.6c gezeigt, entsprechende Systemzustände zugewiesen werden können. Die Ermittlung eines Systemzustandes aus vorliegenden Überwachungsdaten erfolgt durch die Bildung eines Statusvektors, d. h. einer Bestimmung der Position der Überwachungsdaten innerhalb des Eigenschaftsraums (Abbildung 3.6d). Anschließend wird die Distanz des Statusvektors zu den Cluster Zentren berechnet, und der Vektor dem Cluster und somit Systemzustand zugewiesen, der die geringste Entfernung aufweist (Nearest Cluster Center)⁶. Mit diesem Verfahren läßt sich aus vorliegenden Überwachungsdaten ein Systemstatus ableiten, wobei der kontinuierliche Strom von Überwachungsdaten zu einem vom Statusvektor beschriebenen Weg im Feature Space führt (Abbildung 3.6e). Nun ist die alleinige Information über den aktuellen Systemzustand, wie Abbildung 3.6f zeigt, nicht ausreichend. Um eine eindeutige Aussage treffen zu können, ist es notwendig, die Geschichte des Systems, d. h. den vom Statusvektor beschriebenen Weg (die Statussequenz) in Betracht zu ziehen. So ist der in Abbildung 3.6f dargestellte Weg 1 als Verschlechterung des Systemzustandes, der Weg 2 hingegen als eine Verbesserung anzusehen. Um eine derartige Aussage treffen zu können, ist eine Analyse der Statussequenz, d. h. eine syntaktische Mustererkennung erforderlich.

Ziel dieser Methode ist es, bekannte Sequenzen (die Muster) in einer zeitlichen Abfolge von Informationen zu erkennen⁷. Derartige Analysen können mit Hilfe von regelbasierten Systemen oder Automaten durchgeführt werden. Automaten lassen sich durch Zustände sowie Übergänge zwischen diesen beschreiben. Jeder dieser Übergänge wird durch ein bestimmtes äußeres Ereignis, die Eingabe ausgelöst, führt zu dem entsprechenden Wechsel in einen neuen Zustand und leitet eine dem Übergang zugeordnete Aktion, die Ausgabe, ein.

Automaten können durch die in Abbildung 3.7a dargestellten Diagramme repräsentiert werden. Die Kreise bilden die Zustände des Automaten, das erste Symbol am Übergang das diesen auslösende Ereignis und das zweite Symbol die dabei erzeugte Ausgabe. Abbildung 3.7b zeigt beispielhaft die Anwendung eines Automaten zur Analyse der aus obigem Beispiel gewonnenen Statussequenz. Zunächst sind die drei Automatenzustände zu erkennen, welche durch die drei bekannten Systemzustände repräsentiert werden. Die Übergänge werden durch den jeweils aktuell ermittelten Systemzustand ausgelöst, führen zu dem entsprechenden Wechsel in den neuen Automatenzustand und geben dabei den aktuellen Systemstatus aus. Auf diese Weise wird die Unterscheidung der in Abbildung 3.6f dargestellten Wege möglich.

Die in Abschnitt 2.3.3 vorgeschlagenen, regelbasierten Methoden zur Fehlererkennung und Systemkontrolle kommen im Prototypen nicht zum Einsatz, was nachfolgend näher begründet wird. Zur Interpretation eines Systemzustandes (beispielsweise einer Oszillation) ließen sich, wie Abbildung 3.8 zeigt, sowohl Automaten als auch regelbasierte Systeme einsetzen. Bei der Überwachung einer Pufferauslastung sind drei Situationen (Kontexte) zu unterscheiden: normal, kritisch und oszillierend. Bei der Implementation über einen Automaten werden die jeweiligen Kontexte durch die Zustände der Automaten repräsentiert (A_0 : kritisch; A_1, A_8 : kritisch; A_9, A_{10} : oszillierend) und die Übergänge durch Signale für ein Über- (1) oder Unterschreiten (0) des Schwellwertes von 70 % ausgelöst. Die rechte Seite der Abbildung 3.8 zeigt, wie dieselbe Funktionalität mittels eines regelbasierten Systems erreicht werden kann. Hier wird der aktuelle Kontext über eine entsprechende Variable festgelegt und die Kontextwechsel über *if... then*

⁶Bei diesen Berechnungen ist die angesprochene, im einfachsten Fall euklidische Metrik von Vorteil, da sich die Entfernung durch den mehrdimensionalen Pythagoras bestimmen läßt. Bei komplizierteren Metriken sind entsprechende Transformationen erforderlich.

⁷Beispiel ist ein Tresorschloß, wo neben dem Erkennen der Ziffern eine Interpretation der Reihenfolge erforderlich ist, oder Noten aus denen Akkorde und Themen erkannt werden.

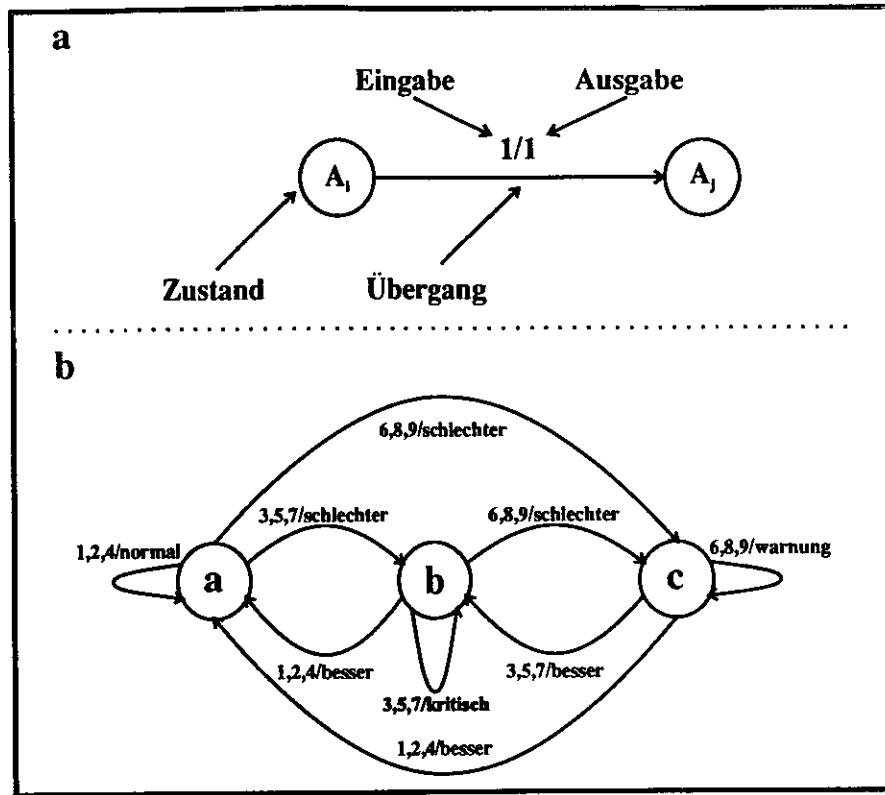


Abbildung 3.7: Abbildung a zeigt die üblicherweise zur Darstellung eines Automaten eingesetzte Notation. Abbildung b gibt ein Beispiel für einen Automaten, der zur Interpretation der in Abbildung 3.6 dargestellten Wege eingesetzt werden könnte.

...-Regeln vorgegeben. Die Vorteile der Automaten liegen in der einfachen Implementation, die mittels einer geläufigen Programmiersprache (C, PASCAL, ...) erfolgen kann. Regelbasierte Systeme erfordern eine entsprechende Programmierumgebung und werden im Interpretermodus ausgeführt, woraus entsprechende Geschwindigkeitseinbußen resultieren. Die Automaten sind im Gegensatz zu den regelbasierten Systemen determiniert, wohingegen deren Vorteil in einer leichten Erweiterbarkeit besteht. So erfordert eine Änderung im zu kontrollierenden System die Hinzufügung einer Reihe von Automatenzuständen und Übergängen, wodurch dessen Komplexität und Unübersichtlichkeit entsprechend zunimmt. In einem regelbasierten System erfordert eine Änderung einzig das Hinzufügen weiterer Regeln.

Der Prototyp beschränkt sich allein auf den Einsatz von Automaten, da in dem momentanen System mit seiner geringen Anzahl von Überwachungsdaten und daraus erkennbaren Fehlern deren Vorteile überwiegen. Die Automaten sind schneller zu implementieren, insbesondere bei den vorliegenden, kleinen Suchbäumen um vieles schneller und bieten den in Echtzeitsystemen

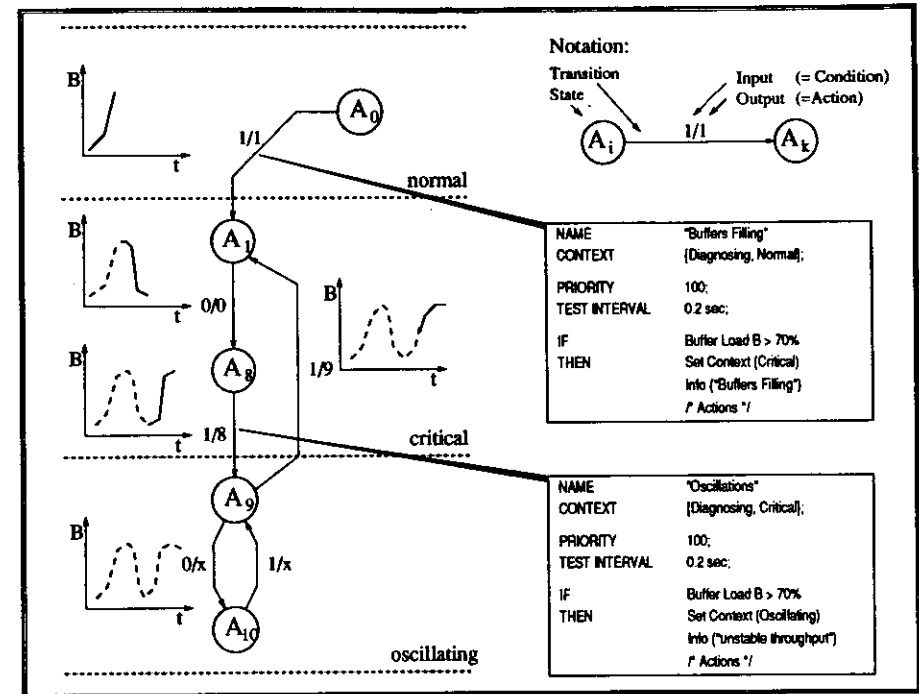


Abbildung 3.8: Die Überwachung einer Größe B auf normale, kritische oder oszillierende Werte. Die linke Hälfte der Abbildung illustriert den im Prototypen gewählten Ansatz eines Automaten. Die rechte Seite zeigt, wie die selbe Funktionalität mittels eines regelbasierten Systems erhalten werden kann (Aus [Beh 93]).

wichtigen Vorteil der Determiniertheit. Zudem muß keine kostenintensive Programmierumgebung erworben werden.

3.3.3 Architektur und Arbeitsweise des Prototypen

Die Architektur des Prototypen ist dem Datenflußdiagramm aus Abbildung 3.9 zu entnehmen. Zentralen Teil des Systems bildet ein Server (Expert System Server), über den der gesamte Kommunikationsfluß sowie der Datentransfer zwischen den Modulen stattfindet. Die Kommunikation der Module mit dem Server erfolgt über TCP/IP-Verbindungen, so daß es ohne weiteres möglich ist, auch auf anderen Rechnern laufende Prozesse zu integrieren.

An den Server sind zwei Prozesse als Datenquellen angebunden. Die Überwachungsdaten können über einen 'Rerun'-Prozeß (rerun) zur Nachstellung alter Runs oder über einen Online-Prozeß (accept online monitoring info) dem Server zugeführt werden. Der Rerun-Prozeß liest auf Datenträger gespeicherte Überwachungsdaten und sendet diese mit einem entsprechend nachgestellten Zeitverhalten an den Server, so daß eine exakte Reproduktion alter Runs möglich ist. Da es auf diese Art möglich war, das System mit aus dem Betrieb des Datennahmesystems

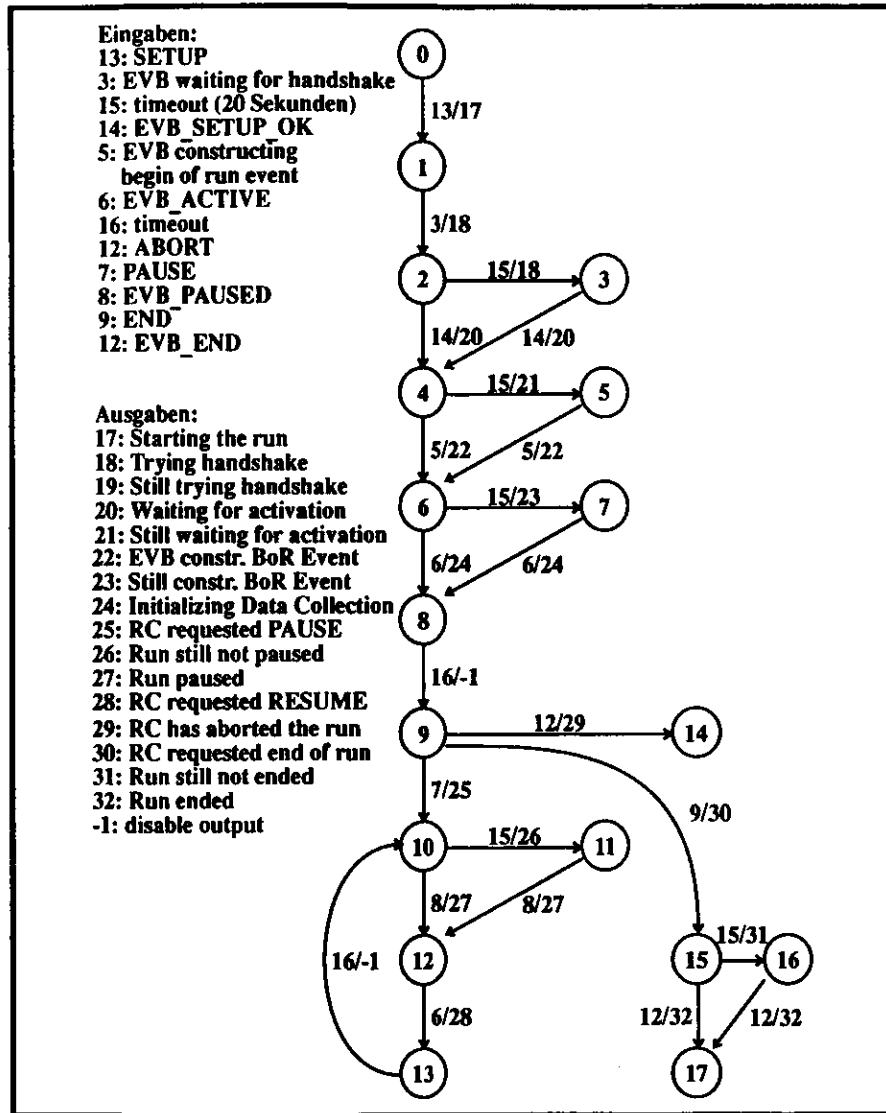


Abbildung 3.10: Der Hauptautomat des Überwachungs-/Diagnosemoduls. Nähere Erläuterungen sind dem Text zu entnehmen.

nur Statusänderungen einem übergeordneten Automaten zuführt. Dieser Automat interpretiert die vorliegende Sequenz und erzeugt entsprechende Meldungen über den Systemzustand oder

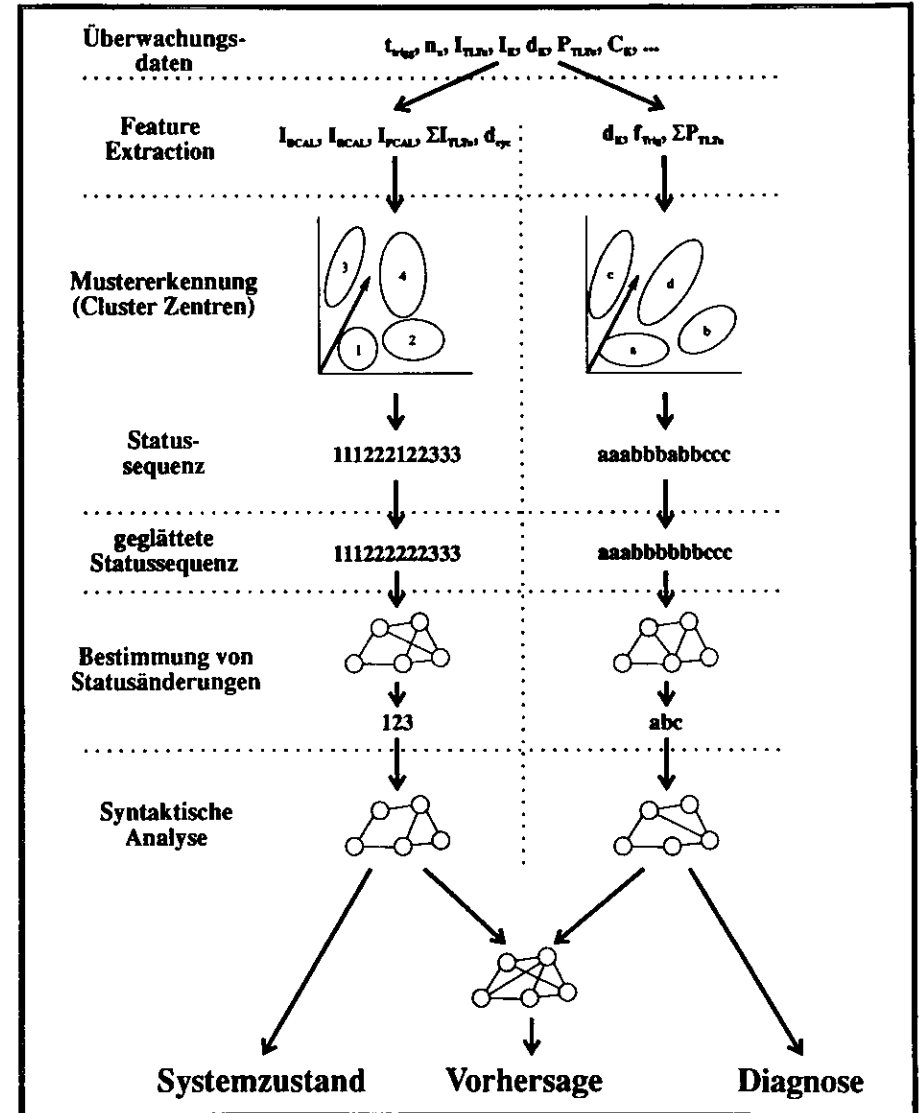


Abbildung 3.11: Die Arbeitsweise des Überwachungs-/Diagnosemoduls. Eine ausführliche Beschreibung findet sich im Text.

die Diagnose. Dabei ist zu beachten, daß der Diagnoseteil nur dann Ausgaben erzeugen kann, wenn der Überwachungsteil einen kritischen Systemzustand erkannt hat. Aus den Meldungen

der Überwachungs- und Diagnose Teile werden Vorhersagen gewonnen, welche sich allerdings auf das Ankündigen einer Systemblockade oder einer möglichen Systemerholung beschränken. Dies erfolgt durch zwei Automaten, je einer zur Vorhersage des Stillstandes und zur Ankündigung der Erholung. Diese Aufspaltung erfolgte allein aufgrund der einfacheren Implementation, denn die Umsetzung dieser Funktionalität allein durch einen Automaten wäre ungleich komplizierter gewesen. Der Automat zur Vorhersage einer Blockade ist in Abbildung 3.12 dargestellt. Der zweite Automat bildet das entsprechende Gegenstück für eine Vorhersage der Systemerholung und ist ähnlich aufgebaut. Diese Automaten werden sowohl durch die Ausgaben des Diagnose- als auch des Überwachungsteils angetrieben. Die Implementation der Automaten erfolgte über eine Matrixdarstellung, wobei die Spalte den momentanen Automatenzustand, die Zeile die aktuelle Automaten eingabe und das zugehörige Matrixelement (ein Vektor!) die Ausgabe des Übergangs und den neu einzunehmenden Automatenzustand darstellt. Die vom Überwachungs-/Diagnosesystem erhaltenen Meldungen über den aktuellen Systemzustand, die Diagnose sowie die Vorhersagen werden über den Server der Benutzeroberfläche zugeführt.

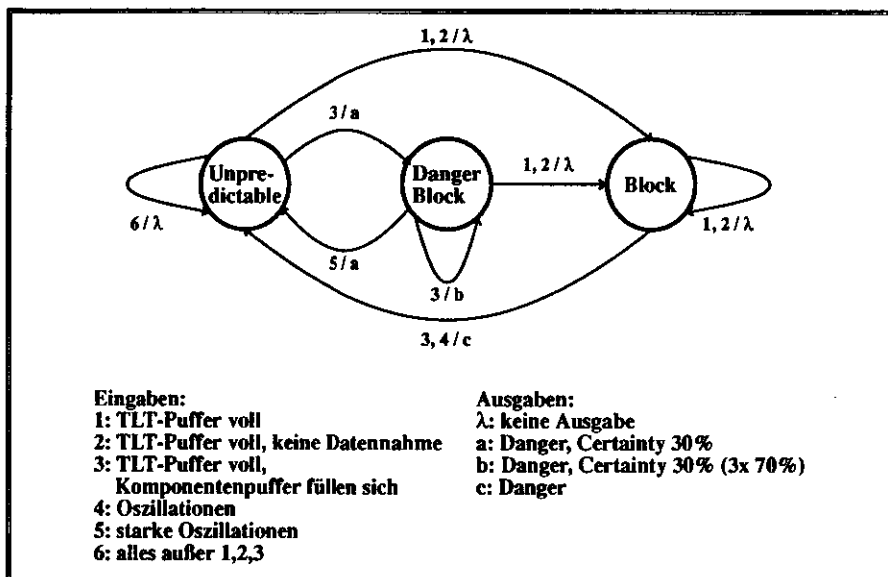


Abbildung 3.12: Der Automat des Überwachungs-/Diagnosesystems der zur Vorhersage einer Systemblockade eingesetzt wird.

Vom Server werden dem auf Bedienung beschränkten Kontrollsystem (ZEX_proposer) die Status-, Diagnose- und Vorhersagemeldungen zur Verfügung gestellt, auf deren Basis anhand eines FSMs die notwendigen Handlungen abgeleitet werden.

Letztes Modul des Prototypen bildet das zur Aktualisierung der Benutzeroberfläche eingesetzte Update-Modul (update ZEX display). Diesem Modul werden über den Server die von den bisher vorgestellten Modulen erzeugten Meldungen zugeführt. Da die Benutzeroberfläche, wie im folgenden Abschnitt näher beschrieben wird, auf dem zentralen Experimentrechner läuft, besteht die Aufgabe des Update-Moduls darin, die aktuellen Daten und Meldungen in

ein entsprechendes Format zu wandeln, und über eine TCP/IP-Verbindung zum zentralen Experimentrechner zu übermitteln. Eine derartige Nachricht wird spätestens nach 300 Überwachungsintervallen oder für den Fall einer aufgrund neuer Status-, Diagnose-, Vorhersage- oder Aktionsmeldungen notwendigen Änderung gesandt.

3.4 Die Benutzeroberfläche

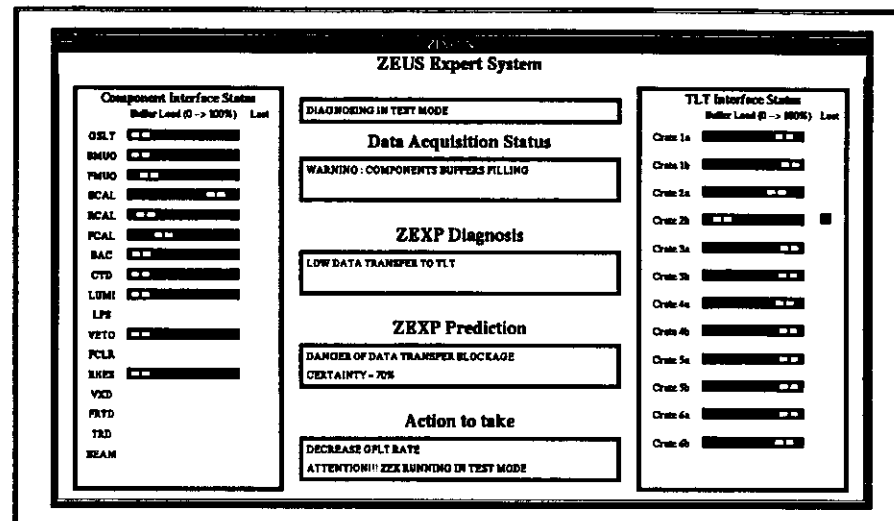


Abbildung 3.13: Die Benutzeroberfläche des Expertensystems. Eine nähere Beschreibung wird im Text vorgenommen.

Die Benutzeroberfläche stellt sich dem Bediener in der in Abbildung 3.13 gezeigten Form dar. Neben den Status-, Diagnose-, Vorhersage- und Aktionsmeldungen werden auch die wichtigsten, die Datennahme beeinflussenden Parameter angezeigt. Dabei handelt es sich um die Pufferauslastungen sowohl auf der Komponenten- als auch der TLT-Seite. Es werden nur die Pufferzustände der aktiven Komponenten angezeigt, so daß auf einen Blick die Runkonfiguration ersichtlich wird. Sowohl für die Komponenten- als auch die TLT-Seite werden die während eines Runs ausfallenden Schnittstellen durch eine entsprechende Markierung in der Spalte 'Lost' gekennzeichnet. Da momentan eine Steuerung des Expertensystems seitens der Benutzer nicht vorgesehen ist, beschränkt sich die Oberfläche lediglich auf eine Anzeige der verfügbaren Informationen. Bei dem der Oberfläche zugrundeliegenden Prozeß handelt es sich um eine, auf dem zentralen Experimentrechner laufende X-Window-Applikation, die unter Zuhilfenahme der OSF/Motif-Programmiersbibliothek erstellt wurde. Die vom Update-Modul des Expertensystems an den zentralen Experimentrechner gesandten Meldungen werden dort von einem Server empfangen und in einem allen auf diesem Rechner laufenden Prozessen verfügbaren Speicherbereich geschrieben. Die Oberflächenapplikation liest die in diesem Speicherbereich abgelegten Daten einmal pro Sekunde aus und nimmt beim Vorliegen neuer Informationen eine

entsprechende Änderung des Fensters vor. Diese Applikation kann von jedem interessierten Nutzer, der über einen Zugang zu dem zentralen Experimentrechner und ein X-Terminal verfügt, aufgerufen und auf seinem Bildschirm angezeigt werden.

Kapitel 4

Erste Betriebserfahrungen mit ZEX-P

Nach einer Entwicklungszeit von etwa einem halben Jahr konnte der in Kapitel 3 vorgestellte Prototyp des Expertensystems Anfang 1993 in Betrieb genommen werden. Die Inbetriebnahme, sowie die während der Entwicklung und des Betriebes gewonnenen Erfahrungen sind Gegenstand dieses Kapitels.

4.1 Der Weg zum Online-System

Anhand der im Herbst 1992 aufgezeichneten Überwachungsdaten, wurden zunächst die bereits in Abschnitt 3.2 geschilderten Untersuchungen durchgeführt und erste Offline-Vorversionen des Prototypen erstellt. Da die Entwicklung des Prototypen während des Wintershutdowns 1992/93 stattfand, standen keine Online-Daten zur Entwicklung des Echtzeitsystems zur Verfügung. Aus diesem Grund wurde ein Programm entworfen, welches die alten, zur Verfügung stehenden Offline-Daten (vergleiche Tabelle 3.1) in die neue Online-Datenstruktur (Tabelle 3.2) umwandelte. Auf diese Art war es möglich, mit Hilfe des in Abschnitt 3.3.3 'Rerun'-Prozesses und den aus der Umwandlung erhaltenen Überwachungsdaten, das Echtzeitverhalten nachzustellen und ein Online-System des Prototypen zu entwickeln, welches dann mit dem Beginn des Experimentierbetriebes im April 1993 zur Verfügung stand. Aufgrund der Entwicklung des Systems in einer nachgestellten Echtzeitumgebung traten bei der Integration des Systems in das ZEUS-Datenerfassungssystem keinerlei Probleme auf. Hier zeigte sich deutlich der Vorteil, das System nicht über Simulationen, sondern über aus dem Betrieb gewonnene und aufgezeichnete Daten zu testen.

Mit dem Beginn der Datennahme ging die Wiederaufnahme und der damit verbundene Testbetrieb des HERA-Speicherrings einher, so daß zunächst keine e-p-Kollisionen zur Verfügung standen. Aus diesem Grund wurden mit dem ZEUS-Datenerfassungssystem zunächst Cosmic- und Testruns durchgeführt, wobei erste Erfahrungen mit dem Online-Expertensystem gewonnen werden konnten. Bei den Cosmic-Runs handelt es sich um Datennahmen, deren Triggerkonfiguration so gewählt wird, daß allein eine Aufzeichnung der ansonsten nur bedingt interessierenden kosmischen Strahlung stattfindet. Diese Runs sind besonders am Anfang der Wiederaufnahme des Experimentierbetriebes und der damit verbundenen Testphase des Datennahmesystems interessant, da die in der kosmischen Strahlung vorwiegend auftretenden Myonen in allen Detektorkomponenten Signale gleicher Intensität hinterlassen und somit hervorragend zum Test des Gesamtsystems geeignet sind. Ein typisches Beispiel der dabei auftretenden Ereignisse ist Abbildung 4.1 zu entnehmen.

Kosmische Schauer entstehen durch den Aufprall hochenergetischer Teilchen (vorwiegend

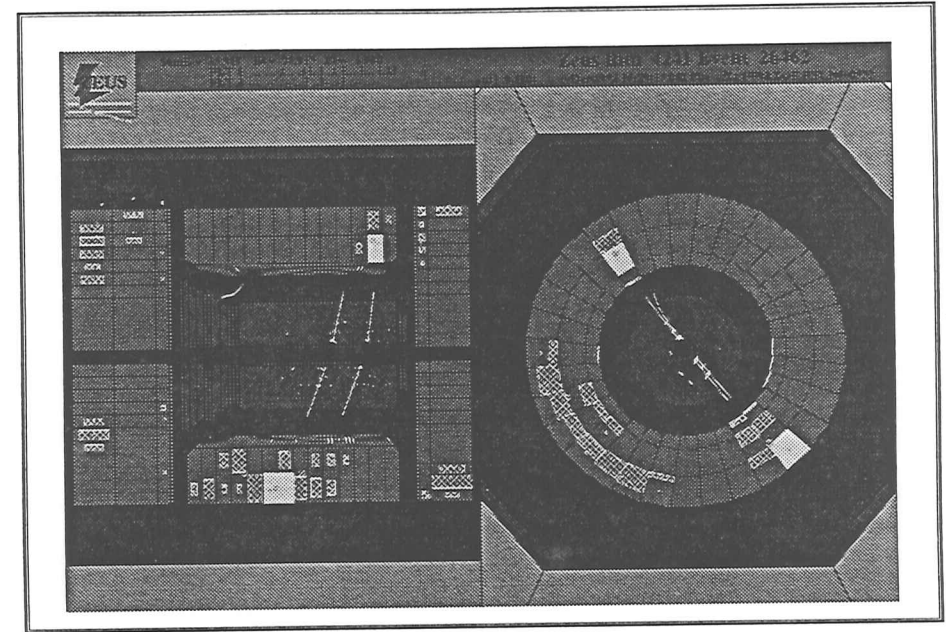
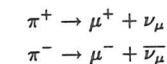


Abbildung 4.1: Ein mit dem ZEUS-Eventdisplay sichtbarmachtes kosmisches Ereignis (nähere Erläuterungen im Text). Dargestellt ist sowohl ein Längs- als auch ein Querschnitt des ZEUS-Detektors.

Protonen) auf die obersten Schichten der Erdatmosphäre. Bei den auftretenden Wechselwirkungen der kosmischen Partikel, den Primärteilchen, mit den Kernen der Luftmoleküle entstehen in großer Anzahl Sekundärteilchen. Dabei handelt es sich vorwiegend um Protonen, Neutronen und Pionen. Da sowohl das Primär- als auch die Sekundärteilchen nach der Kollision noch über beträchtliche Energien verfügen und wiederum auf Luftmoleküle treffen können, wiederholt sich der geschilderte Prozeß kaskadenartig, wodurch sich die Anzahl der Schauerteilchen lawinenartig erhöht.

Die entstandenen, ungeladenen Pionen mit einer Lebensdauer von 10^{-16} s zerfallen schließlich in zwei γ -Quanten. Diese bilden den Ausgangspunkt der elektromagnetischen Komponente des kosmischen Schauers, da durch Paarerzeugung aus den Photonen Elektronen und Positronen gebildet werden. Die entstandenen hochenergetischen Elektronen und Positronen emittieren durch Bremsstrahlung abermals Photonen, so daß sich auch dieser Prozeß lawinenartig fortsetzt.

Die in der hadronischen Schauerkomponente entstandenen geladenen Pionen zerfallen, sofern sie nicht absorbiert werden, nach einer mittleren Lebensdauer von $2 \cdot 10^{-8}$ s gemäß dem folgenden Schema:



Der so entstandene Myonanteil des Schauers erhöht sich mit zunehmendem Alter, bis schließlich auf Meereshöhe 75 % aller Teilchen Myonen sind. Die Ursache liegt in dem deutlich abweichenden Energieabgabeverhalten der Myonen im Vergleich zu anderen Schauerteilchen. Die Lufthülle stellt ein Target von 12 Wechselwirkungslängen und 28 Strahlungslängen dar [Gru 85], so daß Hadronen und elektromagnetische Schauer bis zu einigen 100 GeV absorbiert werden. Myonen hingegen geben nur wenig Energie ab, hauptsächlich durch Ionisation und der Energieverlust erreicht erst oberhalb einiger hundert GeV Myonenenergie infolge von Bremsstrahlung und direkter Paarzeugung eine bedeutsame Größenordnung.

Die kosmischen Myonen liegen energetisch im Bereich des breiten Minimums der Bethe-Bloch-Funktion, d. h. die Myonen verhalten sich annähernd wie minimal ionisierende Teilchen und passieren selbst massive Uran- und Eisenblöcke mit nahezu konstantem dE/dx . Aus diesem Grund sind die Myonen beim Durchqueren des Detektors in allen Komponenten mit nahezu gleicher Energie und gleichem dE/dx nachweisbar und eignen sich hervorragend zur Kalibration einzelner Detektorkomponenten und Tests des Gesamtsystems.

Abbildung 4.1 zeigt ein mit dem ZEUS-Eventdisplay sichtbargemachtes kosmisches Ereignis. Bei dem Run 4241 der mit einer mittleren GSLT-Triggerfrequenz von 12 Hz und einer durchschnittlichen Eventgröße von 12 kBytes lief, trat als Event 26462 ein interessantes kosmisches Ereignis auf. Zwei Myonen durchquerten den Detektor vollständig, hinterließen sowohl in der zentralen Spurkammer als auch dem Barrel Kalorimeter deutliche Signale und deponierten im Kalorimeter eine Gesamtenergie von 35,9 GeV.

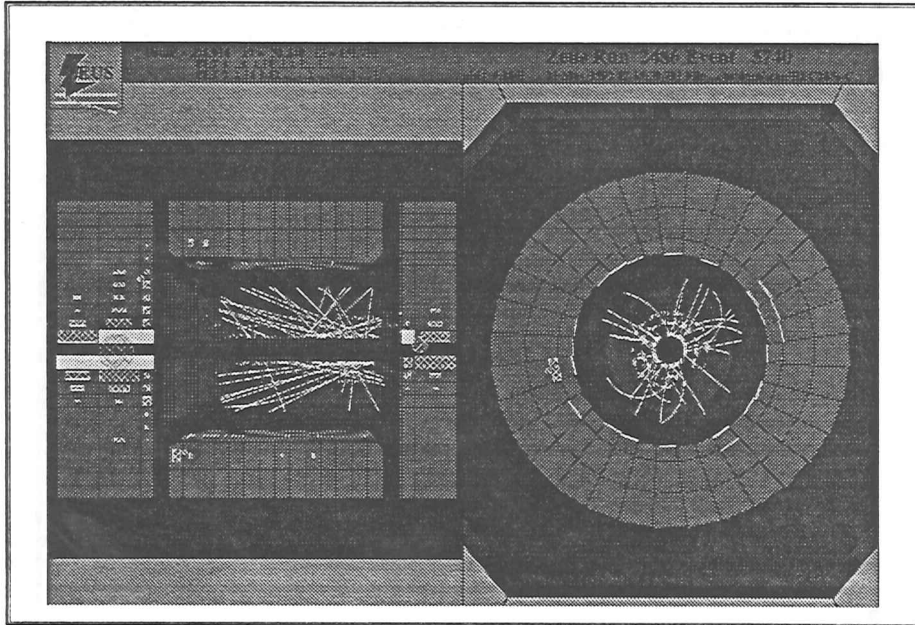


Abbildung 4.2: Ein mit dem ZEUS-Eventdisplay sichtbargemachtes Beam-Gas-Ereignis. (nähere Erläuterungen im Text).

In jedem Run treten eine Vielzahl von Untergrundereignissen auf, deren häufigste Quelle sogenannte Beam-Gas-Reaktionen darstellen. Dabei kommt es zu einer Wechselwirkung zwischen einem Strahlproton und einem Restgasatom im Strahlrohr. Ein typisches derartiges Ereignis, welches während des mit einer mittleren Ereignisgröße von 138 kBytes stattfindenden Runs 2486 auftrat, ist in Abbildung 4.2 dargestellt. Auffällig ist vor allem die hohe Energiedeposition in den Strahlrohrregionen des Vorwärts und Rückwärts-Kalorimeters, sowie die hohe Anzahl von Treffern und rekonstruierten Spuren in der CTD.

4.2 Der Prototyp im Einsatz

Im folgenden werden die ersten Ergebnisse aus dem Betrieb des Prototypen vorgestellt, wobei die Darstellung auf einige typische Beispiele beschränkt bleibt. Bei allen nachfolgend vorgestellten Beispielen handelt es sich um Physik-Runs, die einer Aufzeichnung der e-p-Kollisionen des HERA-Speicherringes dienten. Genauere Informationen zu den einzelnen Runs werden im Zusammenhang mit den jeweiligen Beispielen gegeben.

Während des Testbetriebes konnte der Prototyp den aus den nachgestellten Datennahmen gewonnenen Eindruck bestätigen und arbeitete sowohl präzise als auch zuverlässig. Aufgrund technischer Probleme des zweiten am HERA-Speicherring durchgeführten Experiments (H1) und damit verbundenem eingeschränkten Beschleuniger- und Experimentierbetriebs, werden in den folgenden Beispielen Ergebnisse aus den nachgestellten Runs vorgestellt, welche jedoch auch für das Online-System repräsentativ sind.

4.2.1 Fallstudien

Im folgenden werden typische Situation während der Datennahme und das Verhalten des Expertensystems vorgestellt. Vor dem Starten eines Runs präsentiert sich das Expertensystem mit einem in Abbildung 4.3 dargestellten Bildschirm.

Wichtigste Meldung ist die im Aktionsfenster angezeigte Bestätigung, daß ein Starten des nächsten Runs möglich ist, wobei dies besonders nach einem vorangegangenen, fehlerbedingtem Abbruch der Datennahme von Interesse ist. In diesen Fällen können die Datenpuffer nach wie vor gefüllt und das Starten eines neuen Runs daher nicht möglich sein. Dies ist aber ohne die Hilfe des Expertensystems nur schwer ersichtlich, so daß der Versuch eines Neustarts scheitern kann.

Nach dem Aktivieren des Datenerfassungssystems ist zunächst von allen Komponenten das in [You 91] festgelegte Run Sequencing (vergleiche Abschnitt 1.3) zu durchlaufen. Dabei können eine Reihe von Fehlern auftreten, wobei ein typisches Problem in Abbildung 4.4 dargestellt ist. Hier ist das 'Handshake' des Eventbuilders mit einer der am Run beteiligten Komponente noch nicht erfolgt, so daß keine verlässliche Datenverbindung aufgebaut werden kann. Momentan ist das Expertensystem nur in der Lage, eine allgemeine Meldung über ein Handshakeproblem auszugeben, d. h. es kann nicht erkannt werden, welche Komponente/n den Fehler verursacht hat/haben. Alle während des Run Sequencing vom Expertensystem erkannten Fehler werden von dem in Abbildung 3.10 angeführten Automaten ermittelt.

Die in Abschnitt 3.2 geschilderten typischen Probleme während der Datennahme werden vom Expertensystem zuverlässig erkannt. Abbildung 4.5 zeigt das Beispiel eines Komponentenfehlverhaltens. Der betrachtete Run 4049 lief mit einer mittleren GSLT-Frequenz von 4 Hz, einer mittleren Eventgröße von 119 kBytes und zeichnete insgesamt 81 Events (entsprechend 9,7 MBytes Daten) auf. Während der Datennahme wurden vom GFLT 214, vom GSLT 208 und

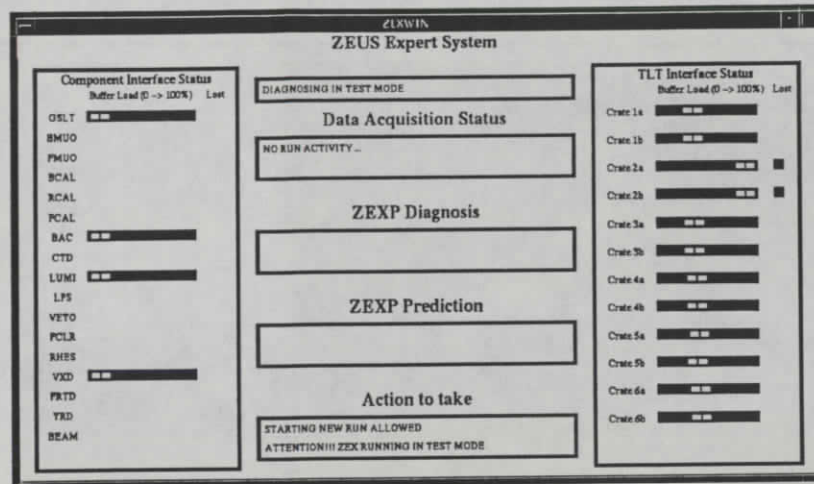


Abbildung 4.3: Die Benutzeroberfläche des Expertensystems vor dem Starten eines Runs.

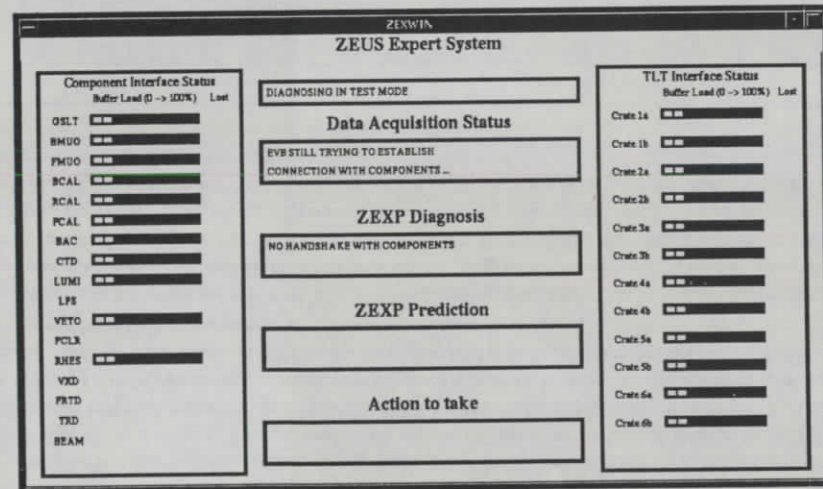


Abbildung 4.4: Die Expertensystembenutzerschnittstelle beim Starten eines Runs.

vom TLT 81 Ereignisse akzeptiert. Etwa 310 Sekunden nach dem Beginn der Datennahme kam es zu einem Ausfall der zentralen Spurkammer (CTD). Dies führte, wie bereits in Kapitel 3.2 ausführlich beschrieben wurde, zu einem Füllen der internen Puffer der TLT-Schnittstelle, was vom Expertensystem korrekt erkannt wird. Der auftretende Bedienhinweis, die Rate zu reduzieren, ist (da dies während der Datennahme nicht möglich ist) so zu verstehen, als daß beim nächsten Run eine andere Triggerkonfiguration zu wählen ist und mit einer entsprechend

niedrigeren Triggerrate gelaufen werden kann.

Der typische Fehler eines Abnahmeproblems auf der TLT-Seite und die Reaktion des Expertensystems sind Abbildung 4.6 zu entnehmen. Während dieses Runs, der mit einer mittleren GSLT-Frequenz von 15 Hz lief, wurden vom GFLT 1880, vom GSLT 1845 und vom TLT 1390 Ereignisse akzeptiert. Bei einer mittleren Ereignisgröße von 116 kBytes wurden 145 MBytes an Daten aufgezeichnet. Während der Datennahme kam es nach 183 Sekunden zu einem Füllen der Datenpuffer. Das zunächst einsetzende Füllen der TPMs, ist, wie in Abschnitt 3.2 näher beschrieben, ein eindeutiger Hinweis auf ein TLT-Abnahmeproblem als Fehlerursache. Daraus resultiert schließlich ein Anstieg der Pufferauslastungen auf der Komponentenseite, was dann zum Stillstand der Datennahme führt.

4.2.2 Eine Systemblockade

Eine interessante Situation ist in Abbildung 4.7 dargestellt. In diesem Run, der mit einer GSLT-Rate von 10 Hz sowie einer mittleren Eventgröße von 153 kBytes lief und 15298 Ereignisse auf Datenträger sicherte, kam es 2855 Sekunden nach dem Aktivieren der Datennahme zu einer Blockade, die ihre Ursache in Transferproblemen auf der TLT-Seite hat. Etwa 2830 Sekunden nach dem Beginn der Datennahme beginnen sich die TPM-Speicher der TLT-Schnittstellen und daraufhin die internen Datenpuffer der TLT-Schnittstellen zu füllen. Dies wird vom Expertensystem wie Abbildung 4.8 zeigt, zuverlässig erkannt.

Nachdem die maximale Füllung der TLT-Pufferspeicher erreicht ist, setzt, da der Eventbuilder keine Komponentendaten mehr zum TLT transferieren kann, ein Anstieg der Pufferauslastung der Komponentenschnittstelle ein. Dies wird vom Expertensystem erkannt und mit einer entsprechenden Meldung (siehe Abbildung 4.9) dem Bediener mitgeteilt. Interessant ist dabei die Vorhersage des Systems, daß eine 70-%ige Wahrscheinlichkeit für eine bevorstehende Systemblockade besteht.

Nachdem auch die Puffer der Komponentenschnittstelle vollständig ausgelastet sind, kommt es zu einem totalen Systemstillstand (Abbildung 4.10). Dabei ist die Diagnose des Systems, eine geringe TLT-Transfergeschwindigkeit mit daraus resultierender Komponentenverzögerung von mehr als 4000 ms interessant. Hieraus ist ersichtlich, daß das Expertensystem beim Vorliegen mehrerer Symptome in der Lage ist, die primäre (die TLT Transfergeschwindigkeit) von der sekundären (die Komponentenverzögerung) Ursache zu unterscheiden.

Nach einem völligen, über etwa 40 Sekunden andauernden Systemstillstand (2855 – 2895 Sekunden nach Beginn der Datennahme), setzt ein geringer Datentransfer ein, was vom Expertensystem mit dem in Abbildung 4.11 dargestellten Fenster bestätigt wird. Hier ist insbesondere die Vorhersage über die Wahrscheinlichkeit der Systemerholung interessant. Im Gegensatz zu einem menschlichen Operator, der von einem völligen Datenstillstand ausgehen muß und eventuell unter erheblichem Zeitaufwand eine neue Datennahme startet, ist das Expertensystem in der Lage zu erkennen, daß nach wie vor ein geringer Datenfluß vorhanden ist. Die Meldung des Systems hilft dem Bediener bei seiner Entscheidung und kann einen zeitaufwendigen Neustart verhindern, was in diesem Fall falsch gewesen wäre da das System die Datennahme fortsetzte. Auch hat sich im Gegensatz zu der Situation vor der Blockade der Bedienvorschlag des Expertensystems in eine Pausierung des Runs geändert. Dies ist sinnvoll, da in der momentanen Situation nicht mehr die GSLT-Frequenz das Problem darstellt, sondern einzig die gefüllten Datenpuffer, die bei einer Pausierung entladen werden.

Nachdem die Puffer sich zu leeren beginnen, das DAQ-System sich von der Überlastung erholt, wird vom Expertensystem eine entsprechende Meldung ausgegeben (Abbildung 4.12).

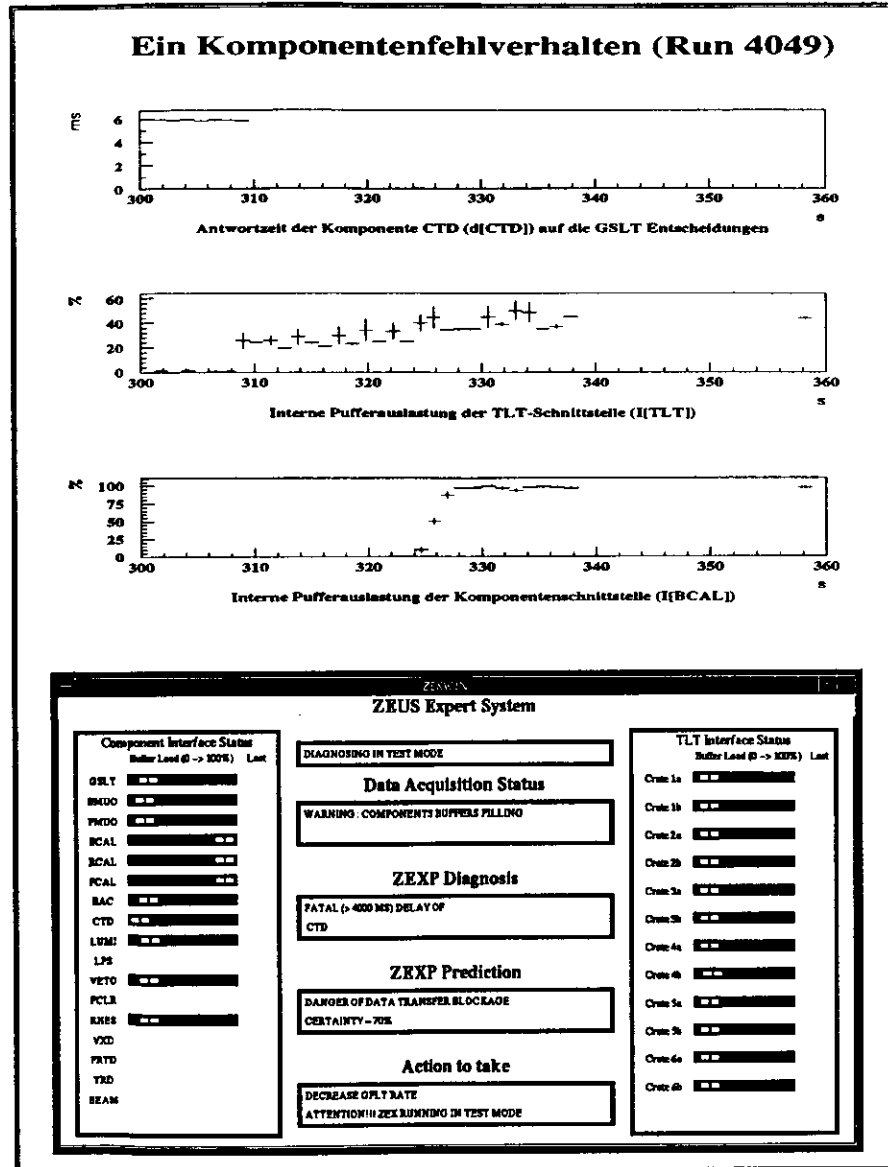


Abbildung 4.5: Ein Komponentengehlerverhalten und die Reaktion des Expertensystems. Nahere Erlauerungen folgen im Text (Vergleiche auch Abbildung 3.4).

Dieser Storfal ist ein eindrucksvolles Beispiel fur die Leistungsfahigkeit des Systems, da

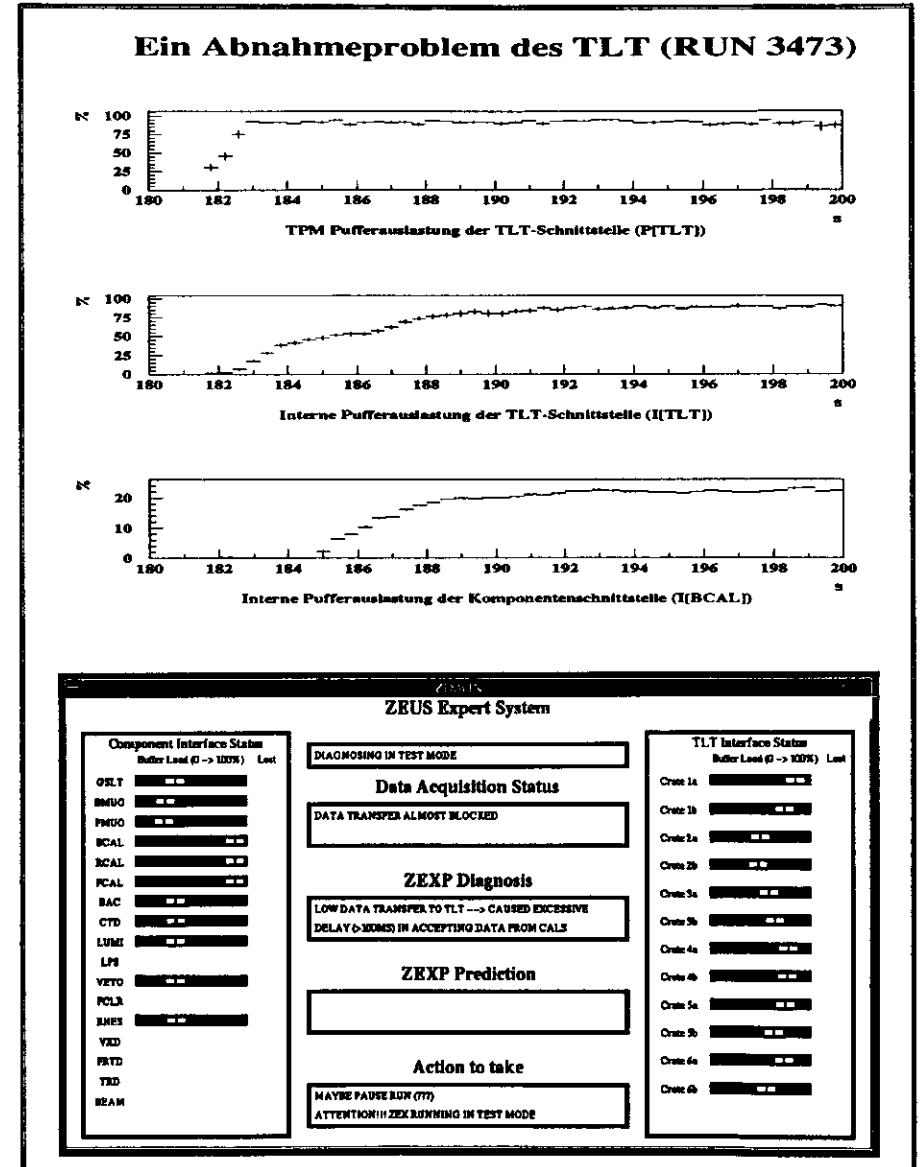


Abbildung 4.6: Ein Abnahmeproblem des TLT und die Reaktion des Expertensystems. Nahere Erlauerungen folgen im Text. (Vergleiche auch Abbildung 3.5)

nicht nur der aktuelle Zustand richtig und zuverlassig erkannt wird, sondern auch sehr gu-

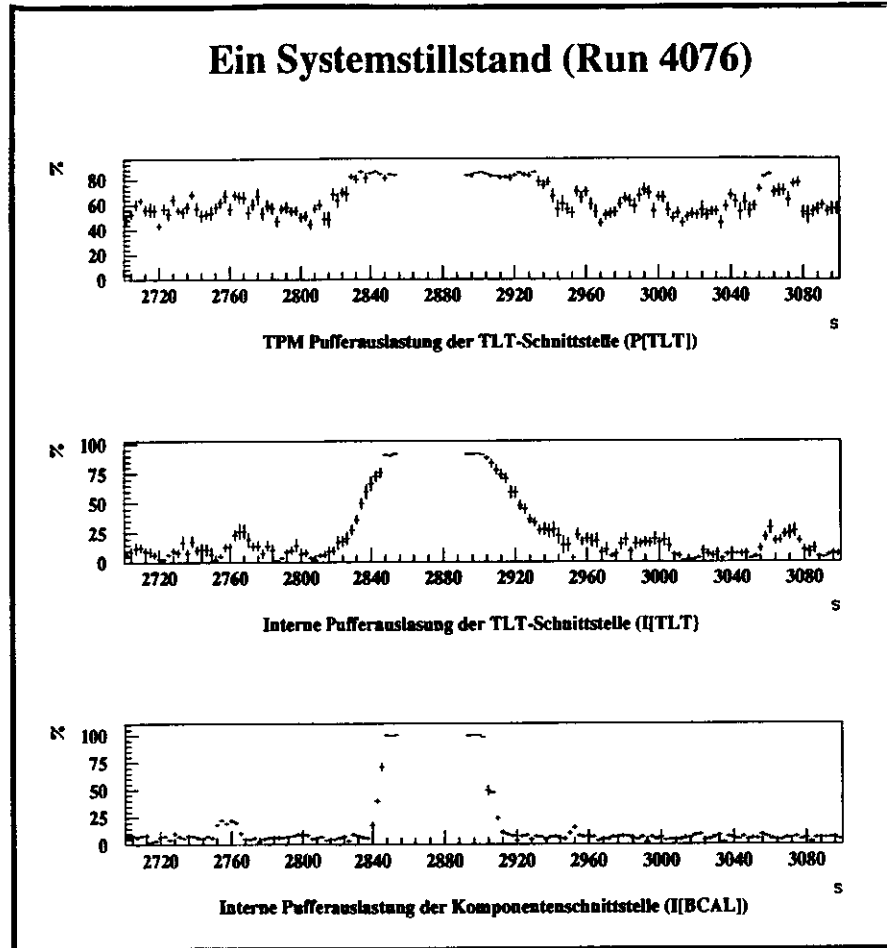


Abbildung 4.7: Eine durch ein Abnahmeproblem des TLT verursachte Systemblockade. (Nähere Erläuterungen im Text.)

te Vorhersagen getroffen werden. Mit diesen und einem in das Expertensystem integrierten Kontrollsystem wäre eine sehr effektive Steuerung des Experiments möglich.

4.2.3 Eine komplizierte Situation und die Vorteile eines ES

Ein weiteres, sehr interessantes Beispiel zur Illustration der Leistungsfähigkeit des Expertensystems stellt der in Abbildung 4.13 angeführte Run 4100 dar. In diesem Run wurden bei einer mittleren Eventgröße von 13 kBytes und einer GSLT-Frequenz von 10 Hz vom GFLT 5687, vom

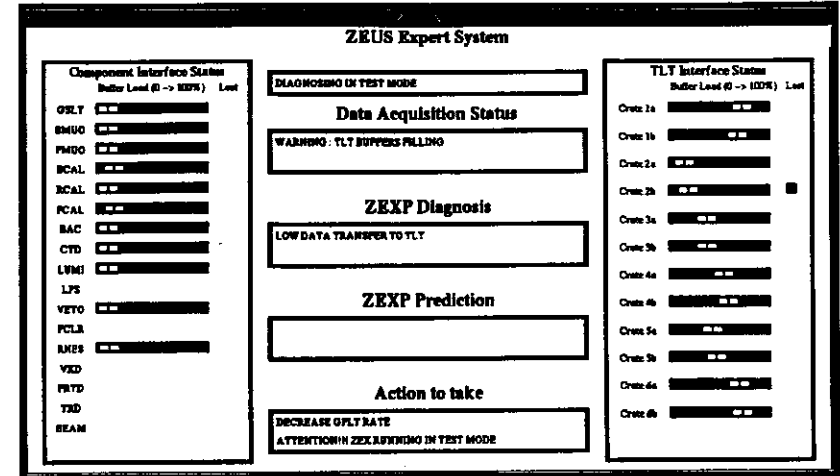


Abbildung 4.8: Die ersten Anzeichen einer möglichen Blockade.

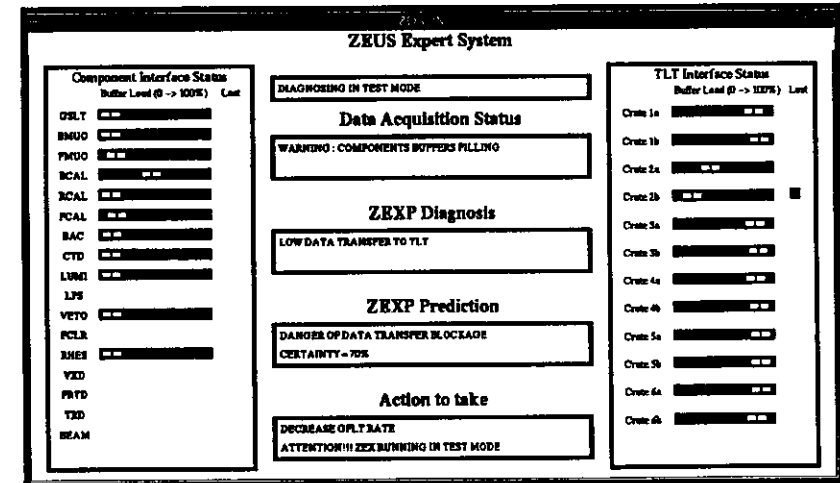


Abbildung 4.9: Die Vorhersage der Blockade.

GSLT 4913 und vom TLT 2928 Ereignisse akzeptiert.

Bei einem ersten Blick auf die vorliegenden Daten liegt die Vermutung eines Komponentenfehlers als Ursache des Anstiegs der TLT-Pufferauslastung nahe, das Expertensystem jedoch meldet Probleme auf der TLT Seite und widerspricht damit zunächst dem Augenschein. Eine umfassende Analyse der in Abbildung 4.14 dargestellten Daten klärt diesen Widerspruch und bestätigt das Expertensystem.

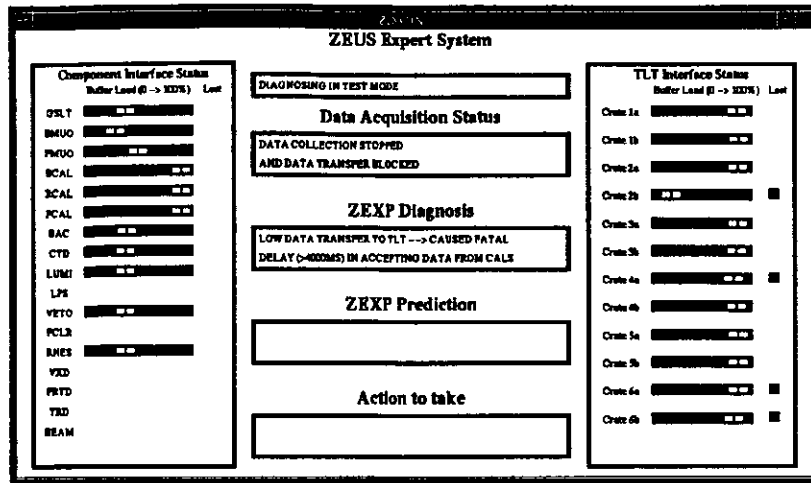


Abbildung 4.10: Die Blockade.

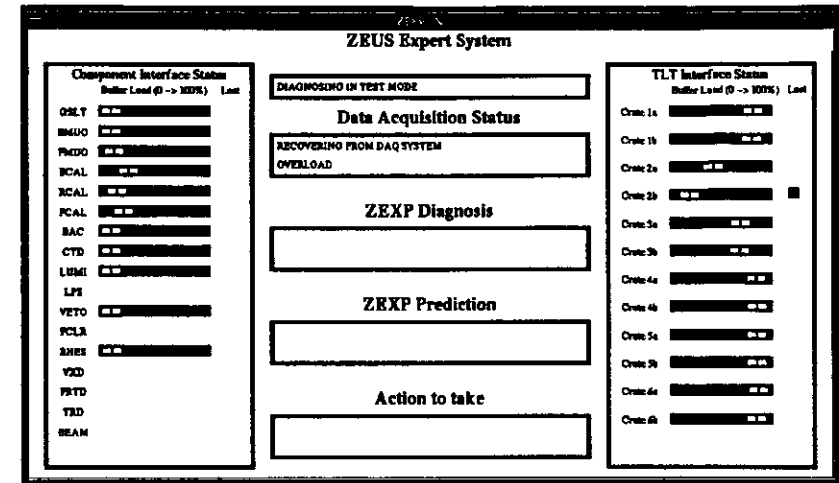


Abbildung 4.12: Die Systemerholung.

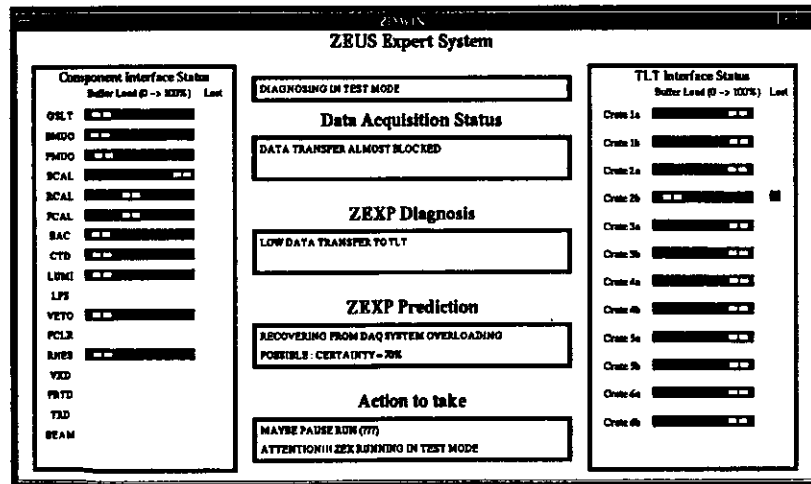


Abbildung 4.11: Die Fortsetzung der Datennahme und die Ankündigung der Systemerholung.

Schon vor der zu betrachtenden Situation, 665 Sekunden nach dem Beginn der Datennahme, trat eine Verzögerung in der Antwortzeit einer Komponente (die zentrale Myonkammer) auf, welche ohne jegliche Folgen blieb (nach 600 Sekunden). Dies läßt bereits vermuten, daß die Komponente nicht die Fehlerursache ist, was durch eine Betrachtung der TLT Übergabepuffer bestätigt wird. Zum betrachteten Zeitpunkt (nach 665 Sekunden) beginnen sich die TPM Puffer der TLT-Schnittstelle zu füllen. Da in diesem Speicher nur vollständige Events abgelegt wer-

den, kann ein Füllen dieser Speicher nicht durch fehlende Daten einer Komponente verursacht werden, so daß die Fehlerursache wirklich, wie vom Expertensystem erkannt, auf eine geringe Transferleistung des TLT zurückzuführen ist.

Dieses Beispiel zeigt deutlich die Vorteile und die Leistungsfähigkeit eines Expertensystems. Im Gegensatz zu einem menschlichen Experten beschränkt sich ein Expertensystem nicht auf die alleinige Betrachtung vermeintlich eindeutiger Daten, sondern operiert stets auf den gesamten zur Verfügung stehenden Daten. Aus diesem Grund besteht für ein Expertensystem eine wesentlich geringere Wahrscheinlichkeit für Mißinterpretationen.

4.3 Möglichkeiten und Grenzen des Prototypen

Seit April 1993 steht die Benutzerschnittstelle des Expertensystems im Kontrollraum des ZEUS-Experiments zur Verfügung und der Prototyp arbeitet als Online-System, d.h. die Daten des Eventbuilders werden in Echtzeit analysiert und entsprechende Meldungen zum Systemzustand der ZEUS-Bedienermannschaft bereitgestellt. Der Prototyp besitzt die von ihm geforderte Funktionalität, d.h. er bietet ein Überwachungs- und Diagnosesystem auf Basis der Eventbuilderüberwachungsdaten. Zudem stellt er ein sehr einfaches Kontrollsystem zur Verfügung, welches sich auf einfache Hinweise an die Bediener beschränkt.

Da der Prototyp lediglich die Überwachungsdaten des Eventbuilders zur Verfügung hat, sind seine Möglichkeiten der Fehlererkennung beschränkt. Das System ist in der Lage, Probleme im Datenfluß zu erkennen und zu lokalisieren, wobei im wesentlichen nur Aussagen darüber gemacht werden können, ob die Fehler auf der Komponenten- oder TLT-Seite lagen. Das System ist nicht in der Lage, die Gründe für die Komponenten- oder TLT-Fehler anzugeben. Dazu wäre es notwendig, umfangreiche Überwachungsdaten der Komponentensubsysteme oder des TLTs zur Verfügung zu haben. Das System kann momentan nur eine Richtung zur Fehlersuche angeben, keineswegs detaillierte Diagnosen vornehmen. Mit den momentan verfügbaren Daten

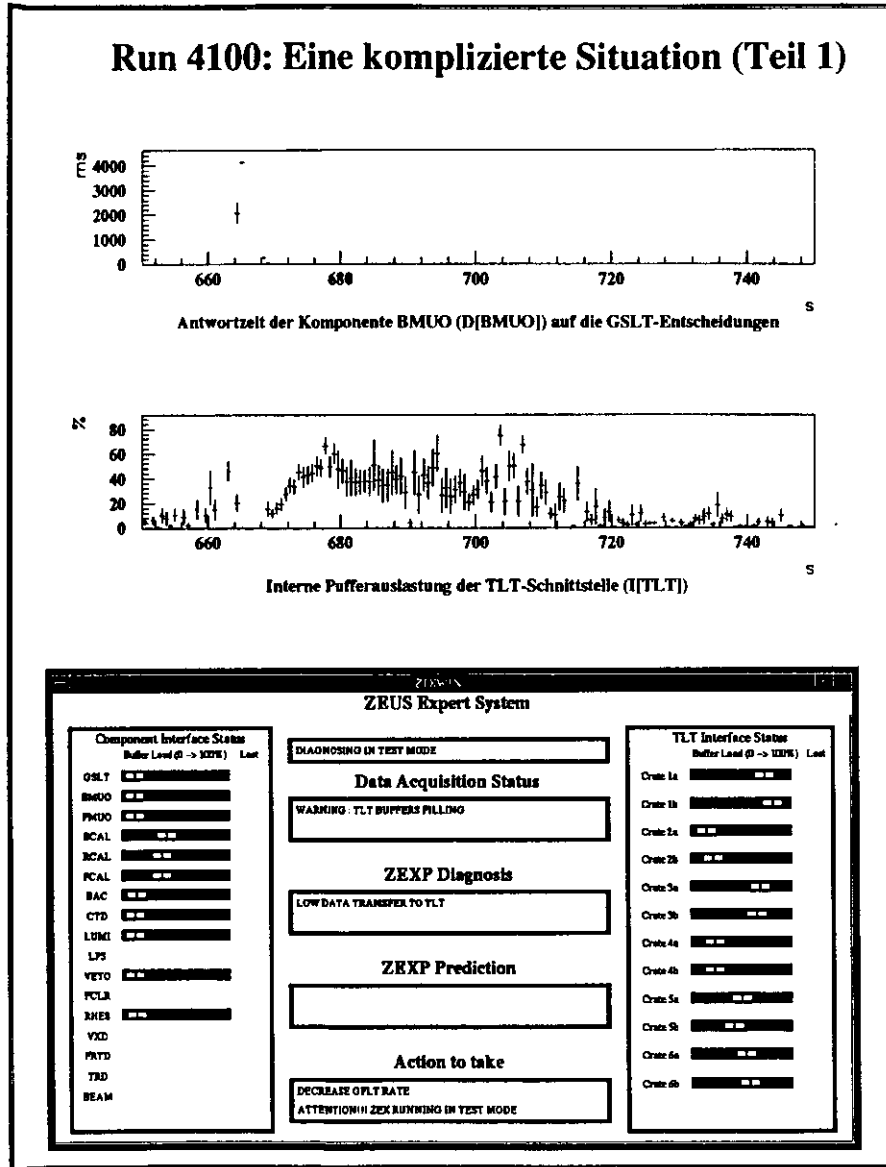


Abbildung 4.13: Eine komplizierte Situation und die Reaktion des Expertensystems. Nähere Erläuterungen im Text.

ist es nicht möglich, Aussagen über die Datenqualität zu treffen. Allerdings ist das System

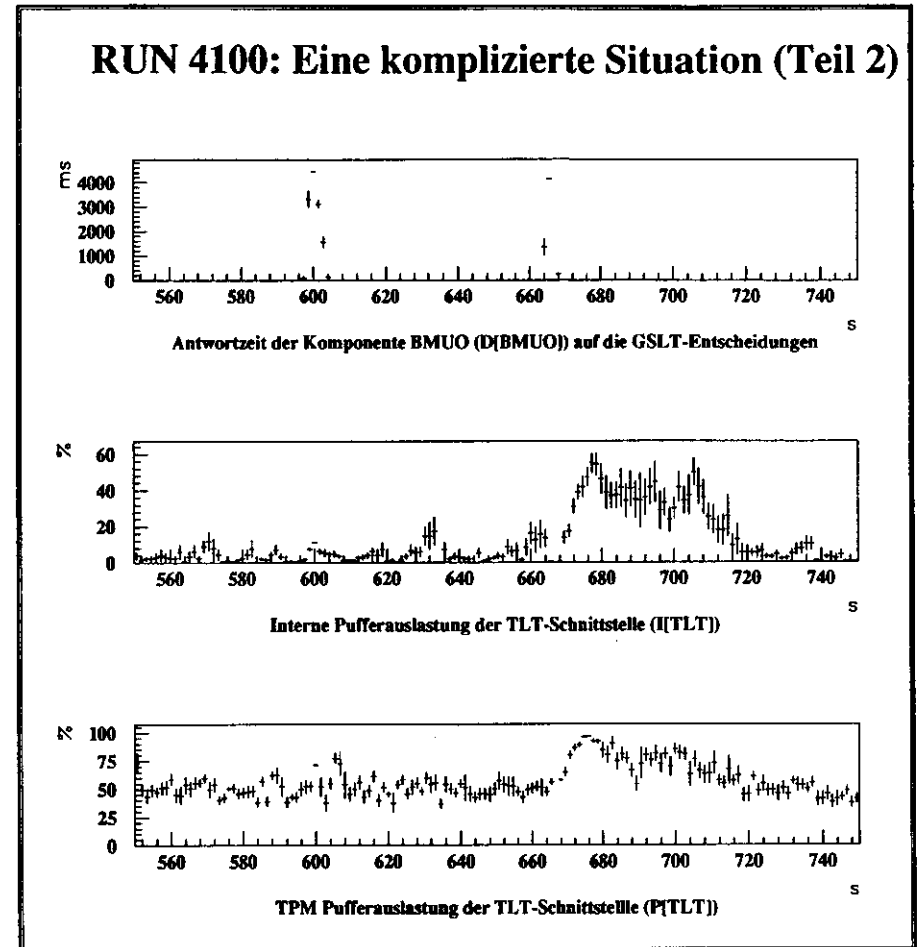


Abbildung 4.14: Ein genaue Analyse der Überwachungsdaten für Run 4100. Nähere Erläuterungen im Text.

in der Lage, alle auf Basis der EVB-Überwachung erkennbaren Probleme zu detektieren und zuverlässige Diagnosen und Vorhersagen zu liefern.

Das im Prototypen verwirklichte, sehr eingeschränkte Kontrollsystem liefert lediglich Hinweise an die Operateure. Die momentane Experimentkonfiguration erlaubt es nicht, Steuerungseingriffe des Expertensystems vornehmen zu lassen, und die Möglichkeiten der Steuerung des ZEUS-Experiments sind momentan sehr begrenzt. Es ist nur möglich, die Datennahme zu starten, anzuhalten oder zu pausieren, jedoch ist es nicht möglich, die Konfiguration oder die Triggerbedingungen während der Datennahme zu ändern.

Für den Prototypen ist kein Lernmechanismus vorgesehen, d. h. das System ist nicht in der Lage, sein Wissen durch Erfahrungen zu erweitern oder zu modifizieren. Dies stellt insbesondere bei einer Änderung der Detektorkonfiguration (beispielsweise höhere Triggerraten) ein Problem dar. Bei einer Verschiebung einzelner Parameter müssen auch die Clusterzentren verschoben werden, was zur Zeit nur seitens des Systementwicklers möglich ist und ein gewisses Fingerspitzengefühl erfordert. Da die zu ändernden Parameter aber in einer externen Datenbank abgelegt sind, ist dies ohne Eingriffe in das System möglich, und die Implementation eines Lernmoduls würde nur eine automatische Änderung der Datenbasis erfordern.

4.4 Der Prototyp als Studie für ZEX

Nach der Vorstellung des Prototypen sollen nun einige Betrachtungen zur Realisierung des angestrebten, in Abschnitt 2 vorgestellten Expertensystems für das ZEUS-Experiment vorgenommen werden.

4.4.1 Realisierbarkeit eines Expertensystems

Die Entwicklung des Prototypen hat gezeigt, daß ein Expertensystem realisierbar ist, wobei insbesondere sehr effektive und verlässliche Überwachungs- und Diagnosesysteme aufzubauen sind. Die Zahl der bereits mit dem Prototypen erkennbaren Fehler ließen sich durch eine Erweiterung des Überwachungsbereiches wesentlich erhöhen.

Problematisch ist die Entwicklung eines Kontrollsystems, wobei dies nicht die Implementation eines derartigen Systems betrifft, sondern die eingeschränkten Möglichkeiten der Experimentkontrolle. Die Kontrollmöglichkeiten bestehen momentan nur darin, einen Run zu stoppen, zu pausieren und neu zu starten. So ist es nicht möglich und erwünscht, die Triggerrate zu verringern, was insbesondere bei kritischen Pufferlasten und einer gleichzeitig hohen Triggerrate eine geeignete Maßnahme wäre. Es ist auch nicht möglich, einzelne Komponenten während eines Runs oder einer Run-Pause zu de- oder aktivieren oder auch neu zu initialisieren. Aus diesen Gründen muß die Effektivität eines Kontrollsystems momentan beschränkt bleiben.

4.4.2 Betrachtungen zur Implementation von ZEX

Bei der Entwicklung des Prototypen erwiesen sich eine Reihe der dortigen Ansätze als erfolgreich und als vielversprechend für den Einsatz bei ZEX. Dabei ist insbesondere der modulare Aufbau des Systems zu erwähnen, der eine einfache Wartung und Erweiterung des Systems gewährleistet. Die Operation des Servers auf TCP/IP-Basis erlaubt die einfache Integration auch von auf anderen Rechnern laufenden Überwachungssystemen. Der wohl erfolgreichste Ansatz war der Einsatz der Automaten. Neben dem Vorteil der Determiniertheit, hohen Geschwindigkeit und einfachen Implementation haben diese Automaten aber bei der Entwicklung des künftigen Systems einen entscheidenden Nachteil: Sie sind äußerst schwer zu erweitern und zu pflegen. Jede Erweiterung des Systems erfordert das Hinzufügen zahlreicher neuer Zustände und Übergänge, so daß es praktisch aussichtslos ist, ZEX wie ZEX-P fast ausschließlich aus Automaten aufzubauen. Stattdessen werden die Automaten zum Erkennen globalen Verhaltens eingesetzt werden können und müssen, denn mit ihnen ist es sehr einfach möglich, Oszillationen oder Tendenzen zu erkennen, was mit anderen Methoden nur schwer zu erreichen ist. Auf den Automaten aufsetzend werden dann die in ZEX-P nicht zum Einsatz gekommenen regelbasierten Systeme Verwendung finden. Zur deren Realisation besteht der Wunsch, hierfür eine kommerziell

verfügbare Entwicklungsumgebung für Expertensysteme einzusetzen. Eine derartige Entwicklungsumgebung böte die entscheidenden Vorteile der leichten Pflege und Erweiterbarkeit, wobei insbesondere die Möglichkeit besteht, das benötigte Wissen der Systemexperten von diesen selber pflegen und implementieren zu lassen. Weiterhin wird es notwendig sein, eine Möglichkeit zur automatischen Justierung der Clusterzentren zu entwickeln, so daß zu deren Wartung kein Experte erforderlich ist.

4.5 Ausblick

Der bisherige Prototyp wird sehr umfangreiche Überarbeitungen erfahren müssen und als nächstes Ziel wird eine Erweiterung des Überwachungsbereichs angestrebt. Hier wird insbesondere an die Integration von Slow Control gedacht, da die dort zur Verfügung stehenden Daten sehr detaillierte Informationen über Zustände der Hardware erlauben. Zudem ist an die Anbindung einzelner Systemkomponenten gedacht, in erster Linie des GFLT, durch dessen Integration es dem Expertensystem möglich sein wird, die Analysen den vorliegenden Triggerbedingungen anzupassen. Dies ist notwendig, da unterschiedliche Triggerbedingungen zu einem leicht differierenden Verhalten des Datenerfassungssystems führt. Das System wird für den Einsatz als Kontrollsystem vorbereitet werden, wobei sich die Kontrolle auf die zuvor beschriebenen Aufgaben beschränken muß. Die Fertigstellung einer ersten Version von ZEX ist für den Sommer 1994 geplant.

Zusammenfassung

In der vorliegenden Arbeit wurde ein Prototyp des zur Überwachung der Datennahme im ZEUS-Experiment geplanten Expertensystems entwickelt und in Betrieb genommen. Das System erkennt kritische Betriebszustände des Datenerfassungssystems und unterstützt die Operateure bei der Datennahme. Das System wurde in einer Betriebsphase Mai-Juni 1993 im ZEUS-Experimentierbetrieb erprobt und liefert bereits aussagekräftige Informationen.

Während der Entwicklung des Systems konnten typische Fehler in der Datennahme erkannt werden. So wurde festgestellt, daß die meisten Probleme aus der am oberen Limit betriebenen TLT-Massenspeicherschnittstelle resultieren. Dies stellt insbesondere für zukünftig mögliche Erhöhungen der Detektorrate ein Problem dar.

Bei der Implementation des Prototypen konnten Erfahrungen für die Entwicklung des endgültigen Systems gewonnen und gezeigt werden, daß der Aufbau eines zuverlässig und in Echtzeit operierenden Expertensystems möglich ist. Bei den gesammelten Erfahrungen ist in erster Linie der erfolgreiche Einsatz der Automaten hervorzuheben. Diese haben sich im Prototypen ausgezeichnet bewährt und werden auch in den folgenden Versionen ihre Einsatzbereiche finden. Allerdings hat sich herausgestellt, daß das endgültige System nicht wie der Prototyp fast ausschließlich aus Automaten aufgebaut werden kann, wobei der Grund in der schwierigen Wartung und Erweiterbarkeit der Automaten liegt.

Der bei der Entwicklung des Systems gewählte Ansatz, das System nicht über Simulationen in Betrieb zu nehmen, sondern stattdessen auf gespeicherte und verfügbare Originaldaten zurückzugreifen und ein Echtzeitverhalten nachzustellen erwies sich als sehr erfolgreich. Der Übergang vom Entwicklungssystem auf ein in das Datenerfassungssystem integriertes Expertensystem erfolgte ohne Probleme.

Die vom Prototyp gelieferten Diagnosen und Vorhersagen sind bereits sehr zuverlässig, jedoch noch nicht detailliert genug, da die verfügbaren Überwachungsdaten des Eventbuilders keine ausreichenden Informationen für genauere Analysen bereitstellen. Dies kann durch die Einbeziehung weiterer Überwachungssysteme erfolgen, die über einfache TCP/IP-Verbindungen an den zentralen Server des Expertensystems angebunden und so in das System integriert werden können.

Literaturverzeichnis

- [ADA 91] S. M. Fisher, P. Palazzi
The ADAMO Data System
Programmers Manual — Version 3.2
CERN/ECP 1991
- [Beh 92] U. Behrens, L. Hagge, T. Schlichting, W. O. Vogel
Status of the ZEUS Eventbuilder
ZEUS-Note 92-054, DESY, Juni 1992
- [Beh92b] U. Behrens, M. Flasiński, L. Hagge
ZEX - Expert System for ZEUS
DESY 92-141 (1992)
- [Beh92c] U. Behrens, M. Flasiński, L. Hagge, W. O. Vogel
The Eventbuilder of the ZEUS Experiment
Proc. Computing in High Energie Physics, 1992
- [Beh92d] U. Behrens, L. Hagge, W. O. Vogel
The Eventbuilder of the ZEUS Experiment
Nucl. Instr. & Meth., 1993. DESY preprint DESY 93-008, 1993
- [Beh 93] U. Behrens, M. Flasiński, L. Hagge, K. Ohrenberg
ZEX — An Expert System for ZEUS
Proc. Eighth Conference on Real-Time Computer Applications in Nuclear, Particle and Plasma Physics, Vancouver, Canada, June 8-11, 1993
- [Bot 93] M. Botlo et. al.
Data Acquisition for Super Colliders
Nuclear Physics B (Proc. Suppl.), 32 (1993) 327-332
- [Gru 85] C. Grupen
Kosmische Strahlung
Physik in unserer Zeit, 3 (1985) 69
- [Neb 88] D. Nebendahl
Expert Systems
Siemens Aktiengesellschaft and John Wiley & Sons Ltd., 1988
- [Pet 89] Wade D. Peterson
The VMEbus Handbook
The VMEbus International Trade Association, Scottsdale, USA, 1989

- [Sch 92] T. Schlichting
Überwachung und Auswertung des Datenflusses im ZEUS "Eventbuilder"
Diplomarbeit am Fachbereich Physik, Universität Hamburg, April 1992
- [Smi 90] W. H. Smith et al.
The ZEUS trigger system
ZEUS-note 89-084, Version 2.0, DESY Hamburg, August 6, 1990
- [You 91] C. Youngman
Run Sequencing
ZEUS-note 91-12, DESY Hamburg, Februar 1991
- [You 89] E. Yourdon
Modern Structured Analysis
Yourdon Press/Prentice-Hall, New York, 1989
- [ZEU 93] ZEUS Collaboration
The ZEUS Detector, Status Report 1993
Technical report DESY, February 1993

Danksagung

An dieser Stelle möchte ich all denen danken, die zum Gelingen dieser Diplomarbeit beigetragen haben.

Mein besonderer Dank gilt Herrn Prof. Dr. E. Lohrmann, der es mir ermöglichte diese abwechslungsreiche und interessante Diplomarbeit im Rahmen des ZEUS-Experiments zu erstellen.

Zudem danke ich Herrn Dr. M. Flasiński für viele hilfreiche Diskussionen und eine Einführung in den Bereich der Expertensysteme.

Ferner möchte ich mich bei Lars Hagge für die stets hilfreiche und ausgezeichnete Betreuung meiner Arbeit bedanken.

Weiterhin danke ich Johannes Mainusch und Ulf Behrens für das angenehme Arbeitsklima innerhalb der Arbeitsgruppe, sowie insbesondere Ulf für viele Mensa-Fahrten.

Schließlich möchte ich mich noch bei meinen Eltern und Tanja für die Unterstützung während des Studiums bedanken.

Erklärung

Ich erkläre, die vorliegende Diplomarbeit unter Verwendung der angegebenen Hilfsmittel und Quellen selbstständig angefertigt zu haben.

Hamburg, im Juli 1993

Kars Ohrenberg