

# Das Kontrollsystem für den Beschleuniger von ANKA

H. Schieler, IK

## Einführung

Das Kontrollsystem dient zur Fernsteuerung und Überwachung einer Vielzahl von Einzelvorgängen innerhalb der Beschleunigeranlage ANKA. Das Kontrollsystem für den Beschleuniger basiert auf Client- und Server-PC's mit dem Betriebssystem Windows NT (Microsoft) sowie dem LonWorks-Feldbus (Echelon) mit intelligenten Knoten und

standardisierten Ein-/Ausgabe-(I/O-Interface-)Modulen, mit denen die einzelnen Geräte direkt an die Server-PC's angebunden werden. Die Server-PC's kommunizieren über CORBA (Common Object Request Broker Architecture) mit den Client-PC's im Kontrollraum. Die gesamte Benutzeroberfläche für die Operateure wurde mit Programmen in der Programmiersprache Java als sog. Java Applets/Applications

realisiert. Der erste reale Test des Systems erfolgte in der Zeit zwischen Oktober 98 und März 99 mit der Steuerung und Überwachung des Vakuumsystems und der Netzgeräte des 53-MeV-Microtrons von ANKA. Im Verlauf des letzten Jahres wurde das Kontrollsystem durch die Einbindung von Diagnostik-Elementen, des Hochfrequenzsystems und der „gerampten“ (d.h. dynamisch dem Strombedarf für ein veränderliches Magnetfeld folgenden) Netzgeräte vervollständigt. Momentan wird das Kontrollsystem weiter optimiert, während es gleichzeitig bei der Inbetriebnahme der gesamten Beschleunigeranlage, bestehend aus dem 53-MeV-Microtron, dem 500-MeV-Booster-Synchrotron und dem 2,5-GeV-Speicherring, erfolgreich eingesetzt wird.

## Die allgemeine Struktur des Kontrollsystems

Die Struktur des Kontrollsystems wurde so geplant, dass das Kontrollsystem aus der Sicht eines Operators so homogen wie möglich erscheint. Das Design des Kontrollsystems erlaubt die Verwendung der gängigen Intranet/Internet-Infrastruktur und WWW-Technologien wie HTML/HTTP, WWW-Browser/-Server mit Java und CORBA/IIOP (Internet InterOrb Protocol). Die Entscheidung hierfür wurde getroffen, weil heutzutage WWW-Browser (z.B. Netscape, Internet Explorer) weit verbreitet sind und weil die WWW-Standards dafür sorgen, dass Informationen aller Art auf den verschiedensten Plattformen gleichermaßen verfügbar und auf einheitlichen Be-

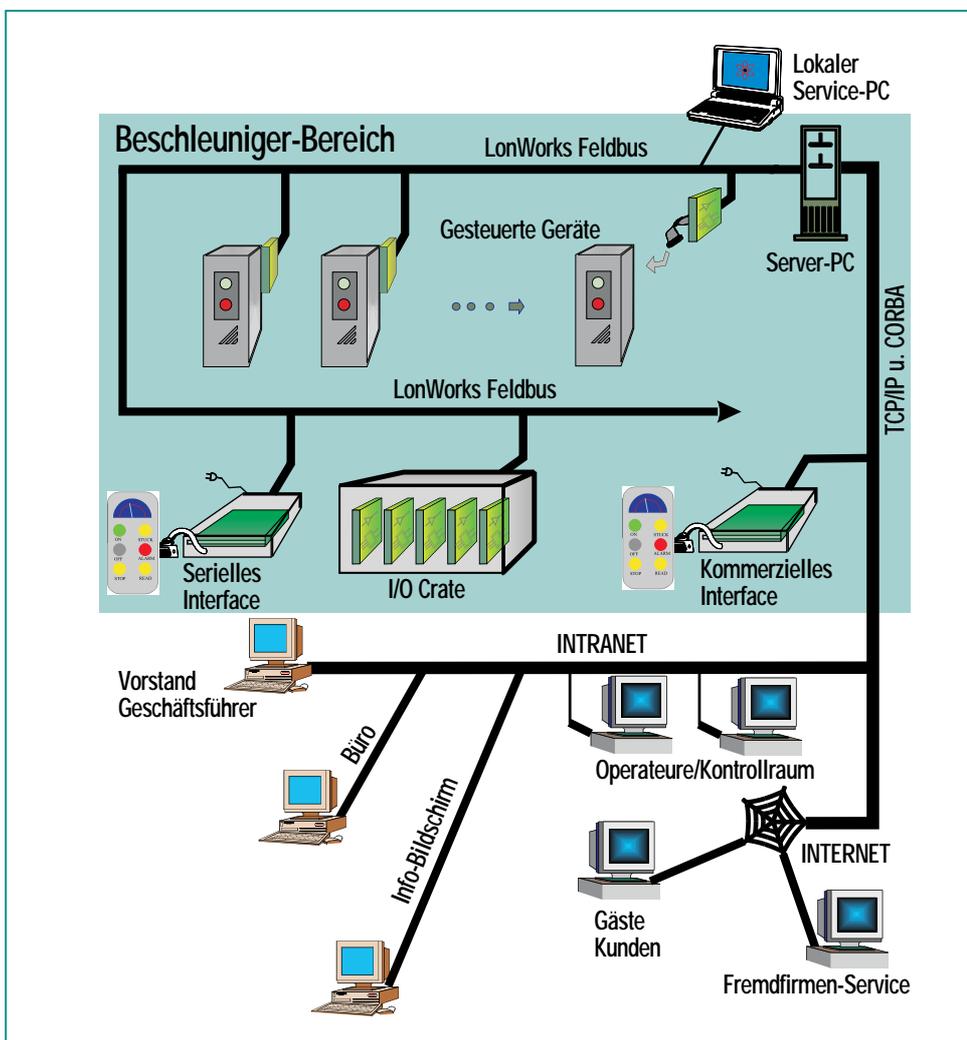


Abb. 1: Schematische Darstellung der allgemeinen Struktur des Kontrollsystems. Der gelb unterlegte Bereich symbolisiert die Visualisierungsschicht, der orange unterlegte Bereich dagegen die Feldbus- und die Prozesssteuerungsschicht.

nutzeroberflächen darstellbar sind.

Technisch gesehen folgt das Kontrollsystem einer 3-Schicht-Standardarchitektur:

- 1) Feldbus-Schicht mit ihrer Geräteanbindung über Schnittstellen
- 2) Prozesssteuerungs-Schicht mit den Beschleuniger-Objekten (z.B. ein Netzgerät)
- 3) Visualisierungs-Schicht mit der graphischen Benutzeroberfläche.

Abstrakt gesehen besteht das Kontrollsystem aus zwei Schichten, die miteinander über die sog. DeviceServer-Programme verbunden sind:

- 1) aus der Feldbus-Schicht mit asynchroner, ereignisgesteuerter Prozessdatenerfassung und Steuerung
- 2) aus der objektorientierten Schicht mit ihrem Modell von Geräten (Devices), in dem die Benutzer/Operateure (Clients) diese ansprechen, als ob diese direkt vor Ihnen stehen.

Abb. 1 zeigt schematisch die allgemeine Struktur des Kontrollsystems.

In den folgenden Abschnitten werden die verschiedenen Schichten etwas genauer beschrieben. Darüber hinaus gehende Details kann man aus [1], [2], [3], [4] entnehmen.

## Die Feldbus-Schicht

Der LonWorks-Feldbus ist ein leistungsstarkes Datenerfassungs- und Netzwerk-System, das bis

32000 intelligente Knoten über I/O-Interface-Module direkt an einen Server-PC mit entsprechendem LonWorks-Feldbus-Interface unter Windows 98/NT anbindet. Das PC-Interface ist mit allen Geräte-I/O-Interface-Modulen durch ein zweiadriges Kabel verbunden. Zur Verwaltung sämtlicher Knoten wird auf dem Server-PC das LonWorks Network Service (LNS) Management Tool verwendet. Die LonWorks-Technologie bietet dem Nutzer ein komplettes Netzwerk-System in Hardware und Software auf einem einzigen Microprozessor (dem „Neuron“ Chip) und erspart diesem somit jegliche Netzwerk-Programmierung in dieser Schicht.

Obwohl viele verschiedene LonWorks-Module kommerziell erhältlich sind, kann es durchaus sinnvoll sein, eigene I/O-Interface-Module auf der Basis des Neuron-Chips zu entwickeln, um individuellen Anforderungen besser gerecht zu werden. Nach einer sorgfältigen Analyse der Anforderungen an das ANKA-Kontrollsystem wurden drei Typen von I/O-Interface-Modulen speziell für ANKA entwickelt und produziert, welche die Anforderungen überwiegend abdecken:

- **Ariadne** ist ein serielles Interface-Modul, das die Standards RS-232, RS-422 und RS-485 mit einer maximalen Baudrate von 115 kbit/s unterstützt. Es hat 16 kByte tiefe Pufferspeicher an den Empfangs- und Sende-Leitungen und eine eigene Stromversorgungs-Einheit für 230-V-AC- (Wechselspannung), unregelmäßige und geregelte 5-V-DC-Stromquellen.

und geregelte 5-V-DC-Stromquellen.

- **Zeus** ist ein hochgenaues I/O-Interface-Modul mit einem 16-Bit breiten ADC (Analog-Digital-Converter) mit 4 analogen Eingangskanälen und einem 16-Bit breiten DAC (Digital-Analog-Converter) mit einem digitalen Eingangskanal, einem DAC-Trigger-Eingang und mittels Optokopplern galvanisch getrennten Digital-Kanälen (8 Eingänge und 8 Ausgänge). Die vier Analog-Kanäle mit einer nominellen Rate von jeweils 1 kHz werden auf den ADC gemultiplext. Bei der Messung wird eine Genauigkeit von 0,3 LSB (Least Significant Bit) erreicht. Der DAC arbeitet mit einer maximalen Rate von 10 kHz. Auf dem Modul befindet sich ein zusätzlicher Microcontroller speziell für die Steuerung der Booster- und Speicherring-Netzgeräte. Dieser stellt einen mit dem DAC-Trigger-Eingang synchronisierten Funktionsgenerator, einen 32 kByte Daten-Speicher für DAC und ADC und einen Selbsttest zur Verfügung.

- **Hera** ist ein reines Digital-I/O-Modul mit 24 Eingängen (50 mA), 8 vordefinierbaren Ein- oder Ausgängen (50 mA) und 8 Ausgängen (solid state relays, 1 A). Sämtliche Ein-/Ausgänge (I/O) sind mittels Optokopplern galvanisch getrennt. Neben dem normalen Betriebsmodus können über die Eingänge auch Frequenzen gemessen werden. Hierzu werden die Eingänge in einem weiteren Betriebsmodus zy-

klisch auf einen internen Frequenzzähler (16-Bit breit; Messbereich 0-100 kHz; absoluter Fehler 1,53 Hz) geleitet.

Alle I/O-Interface-Module wurden mit SMD-Technologie und mit den Abmessungen 160 mm x 100 mm (Europa-Format) gebaut. Sie benötigen eine 5-V-DC-Versorgungsspannung und haben die Ein-/Ausgangs-Verbindungsstecker auf der Rückseite und die LonWorks-Verbindung auf der Frontseite. Abb. 2 zeigt die Module und ihren Einbau am Beispiel eines Schrankes (Racks) mit Netzgeräten.

Die für die Neuron-Chips geschriebene Software, die sog. Neuron-Programme, zur direkten Ansteuerung der Geräte beinhalten recht komplexe Funktionen wie eine Zustandsmaschine und Alarmer, synchrones „Rampen“ in 0.1-ms-Schritten und anderes mehr. Die Kommunikation zwischen Neuron und dem Server-PC erfolgt mittels der LNS-Biblio-

thek mit ihren Netzwerkvariablen und -aufrufen, was gleichzeitig das Management des Feldbus-Netzwerks erlaubt. Auf der Seite des Server-PC's übernehmen die sog. DeviceServer-Programme (siehe folgender Abschnitt) u.a. auch diese Aufgabe. Darüber hinaus werden beim Systemstart sämtliche benötigten Laufzeit-Konstanten aus einer zentralen Datenbank, der sog. Statischen Datenbank, geladen und auf die richtigen Knoten verteilt.

### Die Prozesssteuerungs-Schicht

In der Prozesssteuerungs-Schicht laufen die DeviceServer-Programme, worin die zu steuernden Geräte als Objekte modelliert werden. Diese Objekte (und damit auch die echten Geräte) sind an den entfernt stehenden Client-PC's (z.B. im Kontrollraum) nur über deren in der Software-Schnittstelle offengelegten Parametern zugänglich. Dies be-

deutet praktisch gesehen, dass nur diejenigen Eigenschaften der Geräte dem Nutzer/Operateur zur Einsicht oder Manipulation offengelegt werden, die für die Kontrolle bzw. Steuerung der Anlage erforderlich sind, und dies auch nur in der vom Programmierer vorgesehenen Form (z.B. können sicherheitsrelevante Aktionen nur nach nochmaliger Bestätigung ausgeführt werden). Alle anderen Eigenschaften werden in der Regel vor dem Nutzer/Operateur verborgen gehalten.

Das Accelerator Control Interface (ACI), eine von der Programmiersprache unabhängige Sammlung von Software-Schnittstellen, die als Objekte über das gesamte Netzwerk verteilt sein können, benutzt den CORBA Standard als Netzwerk-Transportprotokoll.

Alle gängigen Beschleunigerkomponenten wie Netzgeräte, Vakuum- und Hochfrequenzkomponenten, Strahlage- und Strahlstrommonitore, usw. sind darin

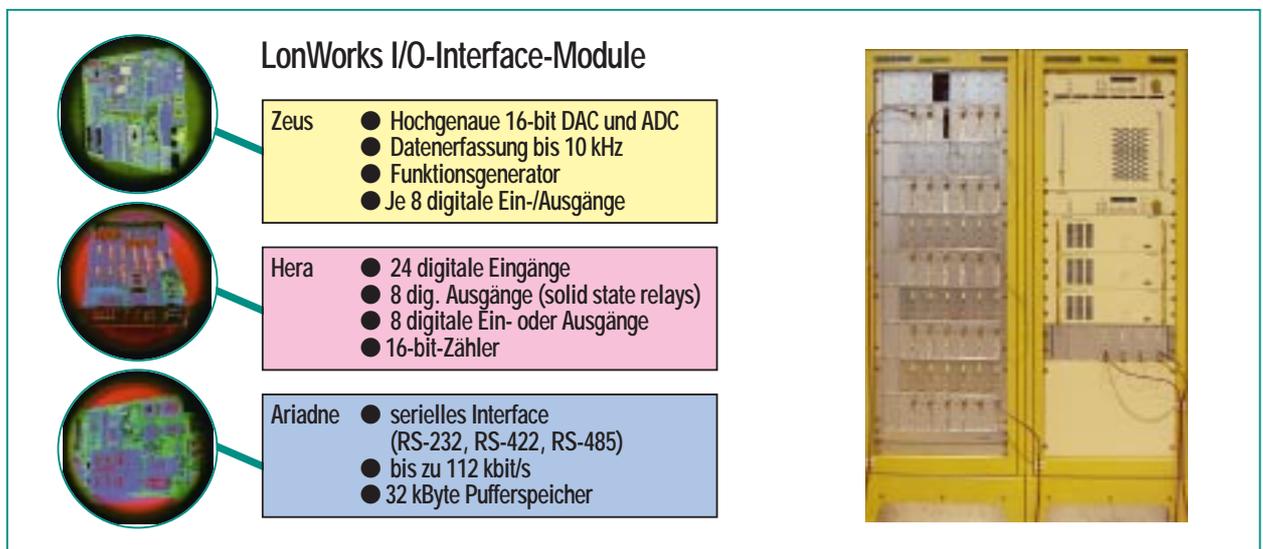


Abb. 2: LonWorks I/O-Interface-Module und ihr Einbau am Beispiel eines der Schränke mit Netzgeräten.

durch ihre Funktionen und ihre Parameter definiert. Jeder kontrollierbare Parameter (z.B. Strom-Sollwert eines Netzgerätes), auch Geräteeigenschaft genannt, ist wiederum ein eigenständiges Objekt, für das Aktionen wie Lesen/Setzen, Inkrementieren/Dekrementieren usw. implementiert sind. Alle mit einer Geräteeigenschaft verbundenen Konstanten wie Minimum-/Maximum-Werte, Gerätenamen und -beschreibung, usw. erhält man direkt aus dem Objekt, d.h. eine Datenbankabfrage ist daher nicht notwendig. Die Parameter werden asynchron über sog. Monitorobjekte aktualisiert

CORBA als Transportprotokoll wurde wegen seiner objektorientierten Struktur und wegen seiner Unabhängigkeit von Betriebssystemen und Programmiersprachen gewählt. Die Geschwindigkeitsanforderungen und die Notwendigkeit mit externen Treibern zu kommunizieren führte zur Wahl von C++ als Programmiersprache für die DeviceServer-Programme auf den Server-PC's.

Die Kommunikation zwischen Benutzer/Operator (Clients) und Geräten (Devices) läuft vollständig asynchron ab. Die Antworten der DeviceServer auf Anfragen des Clients gehen über Rückrufe (Callbacks). Eine andere Art der Callbacks sind die „repeated callbacks“, die oben schon erwähnten Monitore. Hierbei registriert sich ein interessierter Client bei einem oder mehreren DeviceServer, um gewünschte Informationen mit einer bestimmten Aktualisierungsrate automatisch zugestellt zu bekommen.

Vereinfacht ausgedrückt ist die Prozesssteuerungs-Schicht im Wesentlichen ein Protokollumsetzer zwischen dem LonWorks-Feldbus, an dem die Geräte angeschlossen sind und den über das LAN (TCP/IP) und mittels CORBA verbreiteten Objekten, welche die tatsächlichen Geräte für den Benutzer/Operator (Client) modellieren. Hierbei erfüllt diese Schicht auch die Filter- und Verteilfunktion im gesamten System. Abb. 3 zeigt die auf einem Server-PC von 5 verschiedenen DeviceServer bereitgestellten Geräte (exportierten Objekten).

### Die Visualisierungsschicht

Jeder Zugriff eines Benutzers/Operators auf das Kontrollsystem erfolgt über die graphische

Benutzeroberfläche (Graphical User Interface, GUI), welches mit der Programmiersprache Java erstellt wurde und aus sog. „Applets/Applications“ zusammengesetzt ist. In diesen „Applications“ wurden sog. „Java beans“ benutzt, die bei Verwendung geeigneter Entwicklungsumgebungen graphisch, d.h. ohne eine Zeile Programmiercode, erstellt werden können. Diese „beans“ kann man als einzelne Steine eines Baukastens verstehen, die man untereinander zu einem Gesamtgebäude zusammensetzen kann, indem man die von außen zugänglichen Software-Schnittstellen der „beans“ geeignet miteinander verbindet.

Jede „Application“ kann aus zwei Typen von „beans“ bestehen:

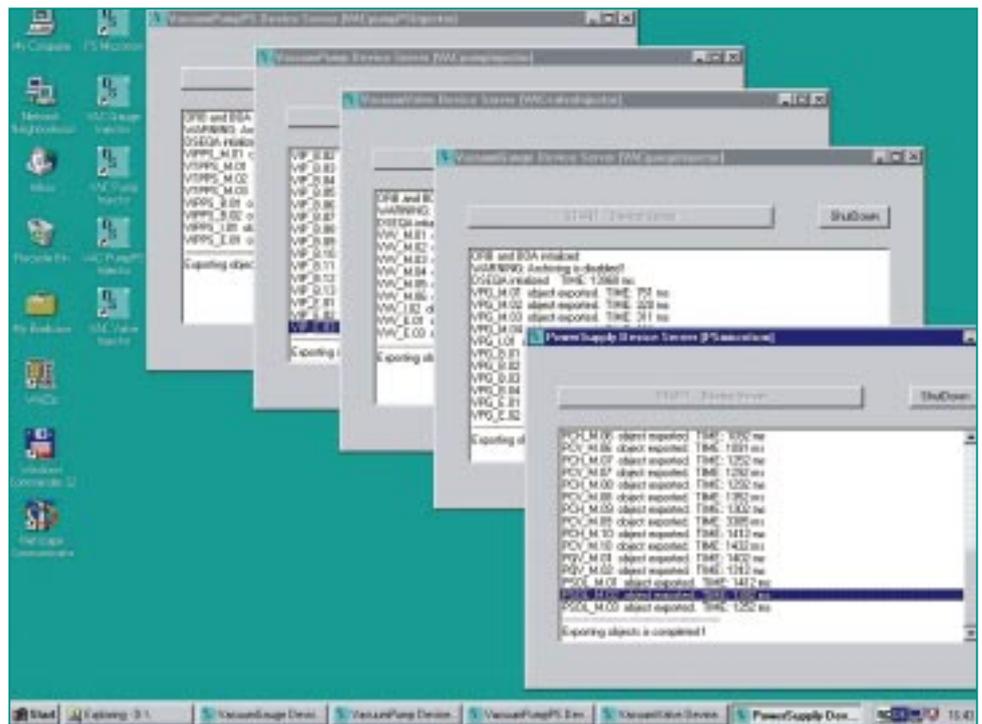


Abb. 3: Bildschirm eines Server-PC's mit den von 5 verschiedenen DeviceServer-Programmen bereitgestellten Geräten (exportierten Objekten).

- „Visual beans“ (GUI Objekte wie Knöpfe, Scrollbars, Anzeigen, Diagramme, ...);
- „Accelerator beans“ (auch „Abeans“ genannt; jede „Abean“ repräsentiert ein reales Beschleuniger-Element, z.B. ein Netzgerät oder eine Vakuum-Pumpe).

Die Kommunikation zwischen Benutzer/Operator am Client-PC und dem DeviceServer auf dem Server-PC der Prozesssteuerungs-Schicht ist in solch einem „Abean“ eingekapselt, d.h. für den Nutzer bleibt das Netzwerk unsichtbar, und damit auch die

Aufgaben der Netzwerk-Kommunikation wie öffnen und schließen einer Verbindung, Übertragung von Daten, Fehlerbehandlung usw.

Während die „Abeans“ selbst erstellt wurden, sind die meisten „Visual beans“ kommerziell erhältlich. Daher ist die wesentliche Arbeit beim Erstellen eines Bedienfeldes für ein Gerät das geeignete Verbinden des entsprechenden „Abeans“ mit den benötigten „Visual beans“ in einer graphischen Entwicklungsumgebung. Abb. 4 zeigt den Bildschirm eines Client-PC's mit drei geöffneten

neten „Applications“ für die Netzgeräte des Injektors von ANKA.

### Das Kontrollsystem im Betrieb

Das Kontrollsystem ist in verschiedenen Ausbaustufen seit Oktober 1998 im Betrieb. In der ersten Stufe wurde mit je einem Client- und einem Server-PC, sowie einem einzigen LonWorks-Feldbus-Strang das 53-MeV-Microtron getestet und in Betrieb genommen. Nach der Einbindung des 500-MeV-Booster-Synchrotrons und des 2,5-GeV-Speicher-

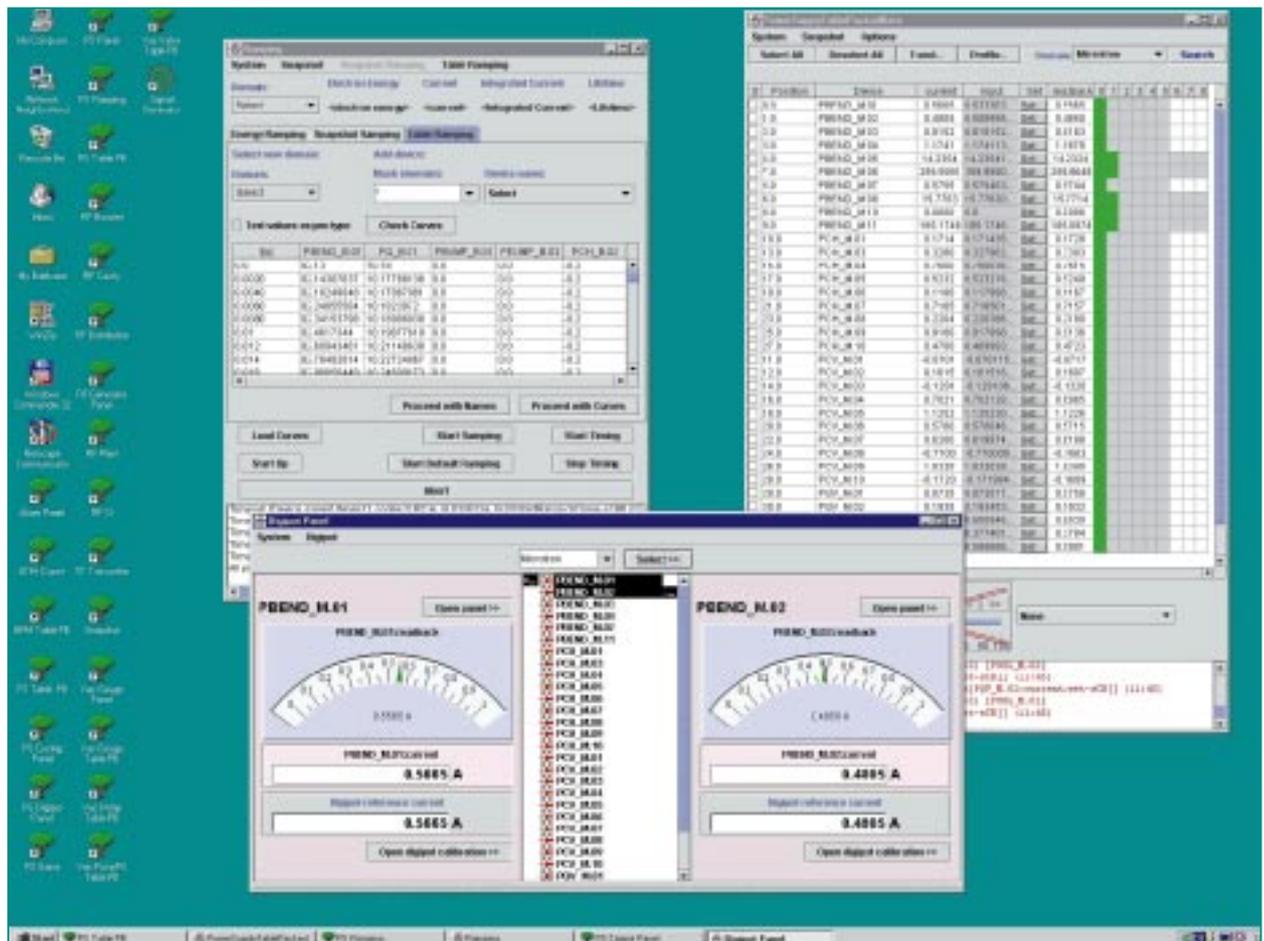


Abb. 4: Bildschirm eines Client-PC's mit drei geöffneten Applikationen für die Netzgeräte des Injektors von ANKA.

rings mit ihren normalen und „gerampften“ Netzgeräten, des Vaku- um- und Hochfrequenzsystems und einigen Diagnostikelementen besteht das Kontrollsystem heute aus folgenden Hardware- und Softwarekomponenten:

- 9 Client- und 9 Server-PC's, alle mit dem Betriebssystem Windows NT 4.0 Workstation mit PII/266MHz bis PIII/500MHz CPU's und 256 MB RAM. Dazu kommen noch etwa 8-10 weitere Rechner mit den verschiedensten Aufgaben (Personensicherheit, Dosimetrie, 20 KV-Anlage, Heizung/Klima/Lüftung, Timing-System, weitere Diagnostik-Elemente, Teststände, Server für die NT-Domäne, usw.), die überwiegend nicht dem hier vorgestellten Kontrollsystem-Konzept zuzurechnen sind. Alle Rechner sind untereinander in einem strukturierten Fast-Ethernet Netzwerk (100 Mbit) über TCP/IP verbunden.
- Auf den Client-PC's läuft die aus Java Applets/Applications bestehende graphische Benutzeroberfläche. Dies sind im wesentlichen Tabellen mit allen Geräten gleichen Typs (z.B.: alle Netzgeräte) oder Schalttafeln, je eine für jedes Gerät. Alle Client-PC's sind gleich konfiguriert und haben jeweils sämtliche Programme verfügbar.
- Die DeviceServer-C++-Programme laufen verteilt auf den 9 Server-PC's. Für jeden Gerätetyp gibt es eine Device-Server-Applikation. Diese fungiert als Software-Interface zwischen dem Client GUI und dem Gerät, indem sie Werte zum Client und Befehle zum Gerät weitergibt.
- DeviceServer-Programme und Client-Applikationen kommunizieren mittels der CORBA-Implementation „Visibroker“ (Inprise).
- Insgesamt 9 LonWorks-Feldbus-Interfacekarten in den 9 Server-PC's sind mit allen 199 I/O-Interface-Modulen (144 Zeus, 39 Ariadne, 16 Hera) durch zweiadrige Kabel verbunden. Dies ist der Feldbus, über den die DeviceServer-Programme mit den Interfacekarten und den zugehörigen Geräten kommunizieren.

Momentan wird das Kontrollsystem weiter optimiert, während es gleichzeitig bei der Inbetriebnahme der gesamten Beschleunigeranlage, bestehend aus dem 53-MeV-Microtron, dem 500-MeV-Booster-Synchrotron und dem 2,5-GeV-Speicherring, erfolgreich eingesetzt wird.

## Literatur

- [1] M. Dach, S. Hunt, B. Jeram, M. Juras, K. Kenda, I. Kriznar, B. Lesjak, K. Mele, T. Milharcic, M. Perko, M. Peternel, U. Platise, M. Plesko, H. Schieler, M. Smolej, R. Sabjan, G. Zkacik, I. Verstovsek, B. Zorko, K. Zagar, *PCaPAC99 workshop, Tsukuba, Japan (January 12-15, 1999)*
- [2] K. Kenda, I. Kriznar, M. Perko, U. Platise, M. Plesko, M. Smolej, *PCaPAC99 workshop, Tsukuba, Japan (January 12-15, 1999)*
- [3] M. Plesko, *PCaPAC99 workshop, Tsukuba, Japan (January 12-15, 1999)*
- [4] M. Dach, S. Hunt, M. Plesko, G. Tkacik, *PCaPAC99 workshop, Tsukuba, Japan (January 12-15, 1999)*