Expertsystem for COSY - Control

N.Bongers, B.Dahmen, R.Danzglock, H.U.Hacker, A.Hardt, C.Henkel, K.Henn,
J.Launer, S.A.Martin, K.Nau, O.Ollech, S.Peischard, A.K.Richert,
H.Singer, K.Sobotta, G.Schug, W.Spieß, A.Weinert, D.Weynen,
C.Röhling (RWTH Aachen)
KFA - Jülich, COSY, Postfach 1913, D-5170 Jülich

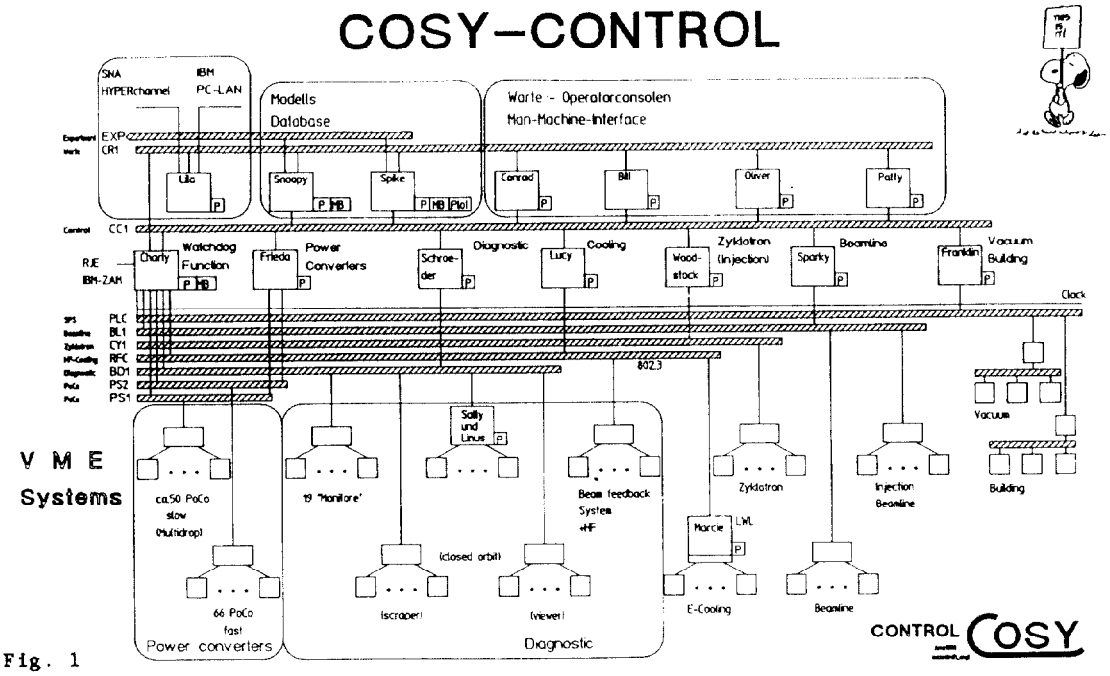

Fig. 1

OVERVIEW COSY-CONTROL-SYSTEM

This is an overview of the COSY-CONTROL-SYSTEM. It is based on several layers with different tasks (Fig. 1).

The first layer is necessary for the interaction between man and machine, the database and the modelling of experiments. The second layer serves the different components of the COSY-MACHINE such as watch-dog-function, power-converters, diagnostic, instrumentation, cooling, vacuum, beam-line and injection. The third layer is for direct process controlling. The connection between the layers are made by a Local Area Network based on an IEEE 802.3 protocol.

The first step in the realisation of this Control-System is to use HP-9000 workstations in connection with HP-9000 minicomputers as workcells. The integration to the peripherie-controller such as power-converters and diagnostic-elements are made by VME-Bus-Crates which are also wired to a LAN based on IEEE 802.3.

A possible hardware solution as far as the VME-Bus integration is concerned is the link between Hewlett Packard Precision Architecture computers (HP-PA) and a VME-Bus interface board named IBAM which is able to drive an IEEE 802.3 protocol (Fig. 2). For the interfacing to the process are several VME-Bus boards with computing-power, memory, digital I/O and function-generators responsible.

The software integration of the VME-Bus is made by a datacom-driver running on the workstations and minicomputers and a link to a monitor-debugger with data communication on a VME-Real-Time Kernel. The several tasks are solved by a dynamic modul reference and a dynamic symbol table which binds a cluster of function moduls (Fig. 3).

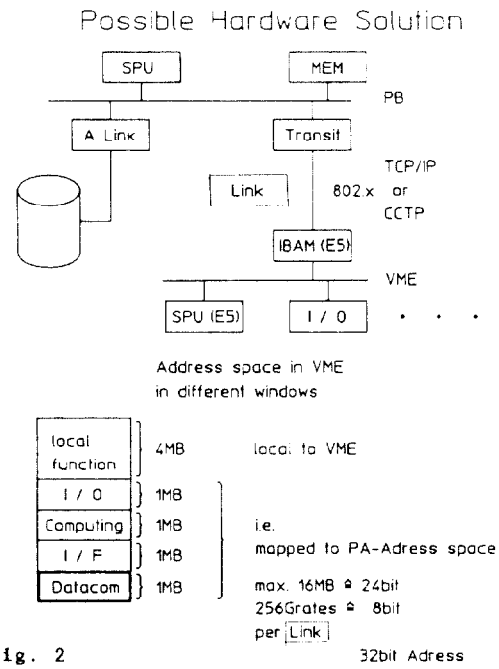

Fig. 2

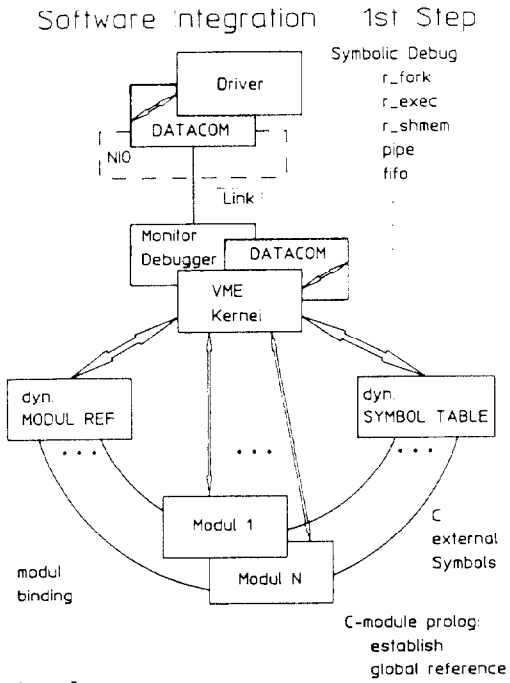

Fig. 3

COSY - Expert - System

To handle and control the complex requirements in an accelerator environment, a knowledge based system will be implemented [1].
The basic model is shown in Fig. 4.

To the analysis of the functions necessary will be done by structured analysis [2].

The definitions and interfaces found will be the input of structured design [4]. This work will be supported by the toolset TEAMWORK [5]. The data flow diagram for the COSY-Expert-System is developed under this method [Fig. 5].

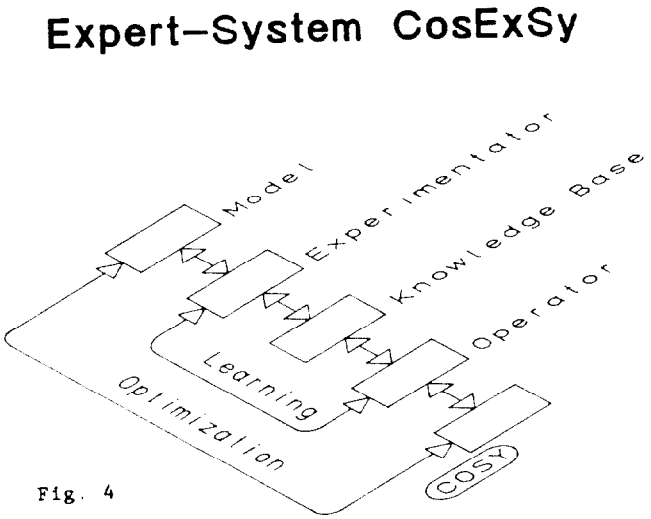

Fig. 4

Man - Machine - Interface

The man-machine-interface will be realized on workstations. The programming environment is built using X-Windows, Starbase graphics and C++ to integrate the Cosy Command Interpreter. This work is performed in a way to connect the knowledge base. The work of knowledge base integration is organised in the tasks following.

- Static Graphic Pictures of Accelerator components
  A hierarchical structure helps the operator to navigate in the system.
  It starts at the overview of the accelerator, and ends at the detailed view of each component with correlating parameters.

- Dynamic Process View
  Historical status of parameters.
  Quasi life display for single parameters connected to an I/O-module.
  This is realized with a transparent transaction concept virtuelized in a X-Window.

- An Interactive Commandprocessor
  This can correct misspelled input or ask for the correct command.
  This tolerante commandprocessor is integrated in the inference structur of the Cosy knowledge base.

This graphic and textual operator surface is also the dialog component of the expertsystem.

Software Integration of heterogeneous Hardware

To integrate all the hardware with one programming environment a special solution is necessary.

The problems are

- an application program will run on different types of hardware in different states of the control system
- accelerator physicists have to develop their own software but nothing to know about the hardware running the application
- integration of accelertor equipment connected to the VME-Bus-Systems and Multidrops as function modules

Our solutions for these problems are

- use of an interpreter instead of compiled modules
- C as language for interpreter
- local / remote stack access
- transaction based communication

resolved in CCdI.
CCdI: COSY-Control distributed Interpreter

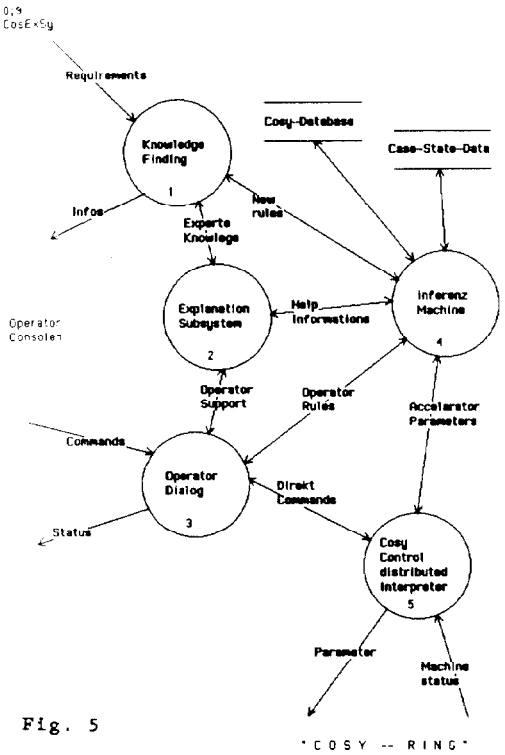

Fig. 5

The Knowledge Base

The major component of the expertsystem is the knowledge base.
Here the major points for the knowledge aquirement

- isolate rules
- delete rules
- modify rules
- Edit
  - -objects
  - -attributes
  - -values
- -modify the instance type

The knowledge base is structured in an external component, supported by a relational database, and an internal component. This internal database is used for the dynamic parameter optimization. The connection between objects and rules and the dialog components are made by a

- inference-machine
- knowledge analyser
- explanation machine

The knowledge base is represented in two ways. First in structural facts which show the rules and objects for solved problems and second a strategic component uses the facts and has the possibility to produce new knowledge.

Distributed Command Interpreter

For the interpretation of commands running the Cosy-Accelerator the Cosy-Control uses a programm package called Cosy-Control distributed Interpreter (CCdI).

The syntax of the interpreter is similar to that of the programming language C. E.g. it is block oriented and allows complex address arithmetic based on pointers. The conception of C is improved by the dynamic linking of procedures. The procedures are declared external to the running interpreter program. The variable types are enlarged by a global external type.

To generate the interpreter standard UN*X tools especially YACC [6] will be used.

Dynamic external procedures and global external variables may be located on any of the computers in the network.

CCdI can schedule distributed tasks in heterogenous systems. So, it integrates various processes over a network.

In a pure UN*X environment, a stackmachine can be accessed by local or remote interpreter. The mechanism to interprocess communication is based on Berkeley IPC sockets. To integrate the process peripherals there is a Function Module interface on the workcells running. The function modules on the VME-Systems can be referenced in this way from the interpreter as functions and data. The communication to the VME-CPU's is based in an transaction oriented message handling system called Cosy-Control transaction protocol (Fig. 7).

Therefore CCdI is a tool for the operator and the accelerator physicist, which provides complex applications in a distributed system by an interactive user interface.

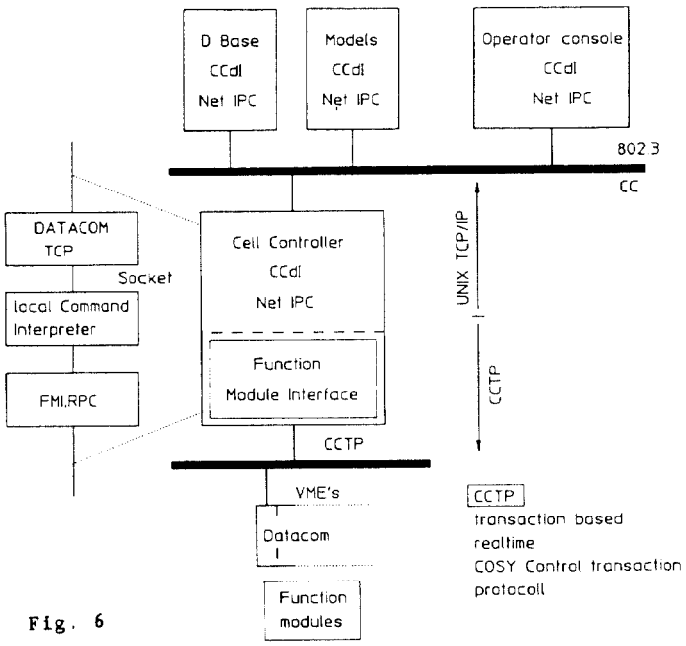

Fig. 6

References

[1] Hofstaedter, Douglas R.,
Goedel, Escher, Bach, 1978

[2] DeMarco, T.,
structured analysis and system specification, 1979

[3] Buchanan, B. & Shortliffe, E.,
Rule-based expert systems, 1984

[4] Page-Jones, M.
The practical guide to structured system design, 1980

[5] Teamwork, CADRE,
management overview, 1987, and program documentation

[6] Johnson, S.C.,
YACC - yet another compiler-compiler, 1975