ARTIFICIAL INTELLIGENCE AND ACCELERATOR CONTROL*

D.P. Weygand
AGS Department, Brookhaven National Laboratory
Associated Universities, Inc., Upton, New York 11973 USA

Introduction

In this paper we review a knowledge-based, domain specific expert system which is under development at Brookhaven National Laboratory to aid in the control of the Heavy Ion Transfer Line (HITL). The expert system is being developed in order to minimize down time after a change of running conditions or at the start of a new run. While the control of HITL is relatively simple compared to a synchrotron, conceptually, many of the problems that are encountered may be extrapolated to more complex machines. We feel that as accelerators become more complex, the need for computer aided control becomes more acute.

Expert Systems and Knowledge Representation

Expert systems in use today span a wide variety of tasks. The DENDRAL system,[1] developed at Stanford, emulates the techniques of chemical experts in analyzing mass spectroscopic data, while MYCIN[2] is adept at diagnosing and treating infectious blood diseases. However, all expert systems have some features in common.

To be considered an expert system, a program must achieve accurate, quality results in a reasonable time. In addition, an expert system must perform its problem evaluation at a high level of abstraction. Truly expert performance generally implies that blind search techniques are avoided to limit the number of hypotheses. Here often an expert system employs many pruning heuristics derived from a human expert. However, one strength of the expert system is that depth of search can be made generally quite deep to achieve difficult results. For example, HITECH, a leading computer chess program consisting of a Sun workstation equipped with specialized hardware, can evaluate 175,000 board positions per second, and examines positions eight levels deep.

However, an expert system is more than just algorithms and performance. A program that numerically solves differential equations has achieved a high level of performance, but does not qualify as an expert system. The essence of an expert system is the ability to reason symbolically. One of the fundamental hypotheses of symbolic reasoning is that knowledge is representational, i.e., knowledge consists mainly of symbolically representing facts about the domain.

Thus, when developing an expert system to aid in beam control at Brookhaven, considerable attention was directed to the representation of knowledge. One must represent the broad range of interrelated information an expert knows about the domain. One needs a symbolic language capable of dynamic storage allocation, just as an expert's knowledge is both symbolic and dynamic. Thus, expert systems are differentiated from high performance, special purpose programs by their use of dynamic, symbolic data representations and symbolic reasoning.

The most important issue which we faced in constructing an expert system for beam control was the representation of knowledge. Beam control knowledge may be divided into three domains. The first is the control segment, i.e., how does one actually change or examine the state of a device. The second segment is the knowledge of how devices effect the beam, i.e., what is a specific device's relationship to the beam. Finally, there are the rules governing how one varies the states of various beam devices to achieve a desired effect or state.

The very lowest level of knowledge is the set of control processes. Within the control processes are embedded all the details of how to control a particular device, i.e., how to change the state of a device or read back the current state of a device. We have developed the control system around groups of controllable entities which we call logical devices. Ultimately, only logical devices are controllable. It is the domain of higher levels of software to create abstractions constructed out of logical devices. We group all knowledge concerning the control of logical devices into the so-called control segment.

Consider a dipole magnet that takes the beam through a 90° bend. The logical device associated with this dipole magnet is the power supply that supplies the current that flow through the coil of the magnet. As far as the control segment is concerned, this device is a power supply just like all other power suplies located about the accelerator. The control segment does not know nor care exactly which magnet (as far as beam optics is concerned) this power supply is serving. Other levels of the control system view this power supply as something more, i.e., a dipole that takes the beam through a 90° bend. The control segment, however, functions merely as a slave, obediently changing the state of logical devices in command from some higher intelligence, be that human operators, specialized applications programs, or an expert system.

There are two types of data within the control segment. One is the software which formats the messages to be sent to devices, and the other is static data about logical devices. Device messages are entirely implemented in C, with a formal, fixed syntax. For example, to change the state of a device, both setpont and/or commands, there is a C routine which takes a device name, a setpoint, and a command state as arguments. Given a device name, the procedure looks up in its static, shared database for relevant information about the particular, named device. For example, the software must know the address of this particular device in the control network, as well as whether the command and/or setpoint being sent is legal for this device. However, this procedure does not understand the command or

*Work performed under the auspices of the U.S. Department of Energy.

CH2387-9/87/0000-0564 $1.00 © IEEE

setpoint in the context of the current goal that is under consideration--it blindly sends legal commands and setpoints oblivious to the wider goal of beam tuning.

We next consider knowledge of accelerator devices at a higher level, that is with regard to the functionality of a device with respect to the beam. The word device now takes a wider meaning--it may not simply be a logical device with a single setpoint, but rather may be a set of logical devices which together make a single, functional device. However, before we consider knowledge of these functional devices, we introduce the concept of a frame,[3] as frames will be used as the representation of higher levels of knowledge.

A frame provides a structured representation of an object or class of objects. (In fact, frames may be used to represent more than just objects and classes, but rather any abstract concept. For example, the rules that form the basis of a rule-based system may themselves be represented as frames.[4]) For example, one frame may represent a specific beam profile monitor (harp) while another frame may represent the entire class of harps. There may be a frame to represent a particular trim dipole magnet, another to represent the entire class of trim dipole magnets, still another to represent all magnets. A frame may have any number of slots for data, and a slot may have any number of facets, and a facet may have any number of values. In the frame representing the specific harp TTL-11MW030 (the names of logical devices are chosen to identify them to human operators; in this case, TTL locates the device in the Tandem Transfer Line, 11 locates it in section 11, MW identifies the device as a multi-wire beam profile monitor, and 30 locates the device approximately 30 feet from the beginning of section 11) resides information particular to this harp, for example, there is a slot which identifies the object called TTL-11MW030 as a harp and another slot for wirespacing. There is no data about how many wires there are in this harp, nor any information about the total width of the harp. If I ask for the number of wires in this harp, I first look in the frame TTL-11MW030 for a slot called NO-OF-WIRES. Finding none, I look at the frame HARP, since TTL-11MW030 is A-KIND-OF-HARP, and "inherit" the value of 16 for the number of wires. If I ask for the width of harp TTL-11MW030, again I look up in its frame for a slot WIDTH. Finding none, I look in the HARP frame again, and find CALCULATE WIDTH in the IF NEEDED facet of the WIDTH slot, which identifies CALCULATE-WIDTH as a procedure to call to calculate the width of a harp. CALCULATE WIDTH then multiplies the number of wires in TTL- 11MW030 (16) by the wire spacing (0.75 mm) and returns 12 mm.

A device's functionality may be represented in its frame, either directly, or inherited through its A-KIND-OF slot. For example, a specific dipole's functionality, say that of TTL-20DH1, may be STEERING, determined from the knowledge that TTL-20DH1 is A-KIND-OF-DIPOLE, however, TTL-20DH1 may also have a functionality of FOCUSING, by virtue of the fact that it may have a large quadrupole moment.

Similarly, frames representing devices which are sensory in nature, must have slots which identify the parameter(s) that a device is sensing. For example, there are processes running at all times which obediently report that a device has fallen out of tolerance, without any comprehension of the consequences of this particular device being out of tolerance. In fact, this process' only task is to inform a cognizant human, i.e., an accelerator operator, that a device is malfunctioning so that the human may assess the significance of the failure, and take appropriate action. The expert system, however, may recognize the impact of a device's malfunction. For example, if an NMR reading from one of the bending dipoles of HITL falls out of tolerance, the expert system may take corrective action by trying to alter the setpoint of the corresponding dipole power supply to bring the NMR reading back into tolerance.

Hence also, this intelligent component of the control system must have information that interrelates and groups devices into logically functional modules. For example, in the case of a beam profile monitor, there are several logical devices which must be activated and read back in order to effect a beam measurement. The expert system must know the logical devices associated with this harp and their relationships. In addition, the expert system must know the logical devices that control the harp's amplifier gains and integration times. The expert system must know where the harp is in the transfer line, and the positional relationship to other beam elements, such as steering and focusing magnets.

Heavy Ion Transfer Line Control Procedure

We now examine the specific procedure that human operators follow to tune and otherwise control and maintain the Heavy Ion Transfer Line, and we shall attempt to recast these rules as a control procedure for an expert system. First it is necessary to have some understanding of the structure of the transfer line.

The beam line is divided into 13 sections, numbered 10 to 23, section 10 being at the exit of the Tandem Van de Graaff, and section 23 being at the entrance to the Alternating Gradient Synchrotron (AGS). The transfer line contains two 90° bends in section 11, two 24° bends in sections 17 and 18, and two 69° bends in section 22. The rest of the line is straight sections to cover the 600 meters between the Tandem Van de Graaff and the AGS.

Active beam elements consist of the aforementioned bending dipoles, as well as quadrupole focusing magnets, and trim dipole magnets both for fine steering of the beam and to counteract the effect of the earth's magnetic field. There are xy trim dipole steerers at all foci and at all quadrupole lens groups. The beam is measured by Faraday cups and beam profile monitors (harps) at all foci.

Beam tuning is performed section by section. Each section, in general, is defined by the elements between foci. We consider the tuning procedure for a typical straight section of the beam line.

A typical section consists of an xy steerer at the upstream focus, a quadrupole doublet with a nearby xy steerer, and a Faraday cup and harp at the downstream focus. The upstream xy steerer is adjusted to aim the beam precisely at the center of the quadrupole doublet. One determines that the beam is hitting the center of the quadrupoles by changing the quadrupole setting--if the beam is centered, the only effect should be on the beam spot size, not the spot position. After the beam is centered, the value of the quadrupole doublet is adjusted to give the correct spot size.

Prior to tuning, a beam tuning program (such as Optic II) is used to give initial values for all bending dipoles and quadrupoles. It is hoped that such a program will initialize values to within 1%.

A simplified view of the goal tree for tuning the beam through section 13 is shown in Figure 1. A subgoal of this goal is that the beam must first be tuned through section 12. In addition, another subgoal is that the beam must be tuned in section 13. The horizontal line through lines to the subgoals indicates that the current goal is achieved only if both subgoal 1 and subgoal 2 are both achieved. (Absence of a horizontal line indicates that subgoal 1 or subgoal 2 must be achieved.) Arrows indicate the order of solution.

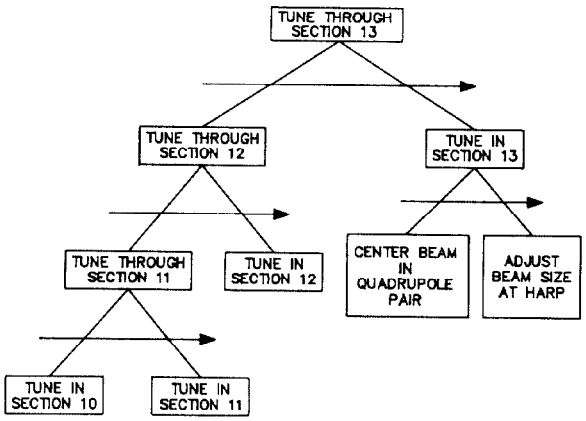

Fig. 1. Simplified view of goal tree.

The Architecture of the Expert System

The control system has been developed on a network of Apollo workstations. The code for communicating with logical devices has been developed entirely in C. There exists a database, implemented as a shared memory file which is mapped into the virtual address space of each control process. This file is called the Device Definition File (DDF). Overlaid over this file is an array of C structures which contain information about each logical device, particularly network addresses. Thus, each control process has access to this database via virtual memory.

The expert system is implemented entirely in interpreted Portable Standard Lisp (PSL). Its database is implemented as dynamic frames. The library of compiled C programs is mapped into the virtual address space of the Lisp interpreter, together with the DDF. Thus, the Lisp interpreter has access to compiled control functions as well as the static DDF. All low-level control functions are thus executed from the C library using the DDF, all high level functions are implemented in Lisp using the frame database.

In addition, the expert system will start up slave processes from time to time, as needed, to assist in certain tasks. For example, there may be a display process begun so that a human operator may see graphically the data that the expert system is basing its decisions. The expert system may also start up a fitting procedure on another computer to do numerically intensive processing. After starting up this process, the expert system then serves as a remote subroutine call to the fitting procedure, changing magnet settings as the fitting program varies fit parameters, and returning data about the state of the beam.

Separate from the frame representation of knowledge, yet central to the functioning of the expert system, is the goal-solving mechanism of the program. This function takes as an argument a specified high level goal, and then, by developing a tree of subgoals, attempts to solve the given goal via a hill-climbing technique. Examples of possible goals are as follows:

POSITION BEAM IN HARP TTL-17MW240 AT 0.5 MM
CENTER BEAM IN QUADRUPOLE TTL-17QH1
TUNE BEAM IN SECTION-17
TUNE HITL BEAM

The earlier goals on the above list may, in fact, also be subgoals of the later goals.

The goal-solving is structured in three layers--the GOAL-MASTER, the GOAL-SUBMASTER, and the GOAL-SOLVER. The GOAL-MASTER establishes some global variables for use during the goal solving procedure, and then calls the GOAL-SUBMASTER. GOAL-SUBMASTER examines the goal that it has been given, and tries to organize subgoals for maximum efficiency. The actual resolution of goals is done by the GOAL-SOLVER. GOAL-SOLVER follows the necessary procedure to achieve the desired goal. GOAL-SOLVER looks for the conjunction and disjunction of subgoals, and recurses on itself appropriately. That is,

(GOAL-SOLVER (A AND B)) = (GOAL-SOLVER A) AND (GOAL-SOLVER B)

While the GOAL-MASTER is called only once for a given goal, the solution of a goal may require substitution with a combination of subgoals, followed by recurssion of the GOAL-SUBMASTER or GOAL-SOLVER. For example, a subgoal of beam tuning may be RECEIVE REPORT FROM HARP TTL-11MW060. If ths goal fails because network communications have broken down, then this subgoal may be replaced by (ESTABLISH NETWORK COMMUNICATIONS) AND (RECEIVE REPORT FROM HARP TTL-11MW060).

Conclusion

While the expert system is currently only under development, it is now able to accept some of the more mundane tasks of beam control. It is beginning to take over archiving functions, and is capable of some simple tuning. In addition, it is beginning to incorporate accelerator modelling programs into its repertoire with the intent of applying its expertise to synchrotron control.

References

1. R.K. Lindsay, B.G. Buchanan, E.A. Feigenbaum, and J. Lederberg. Applications of Artificial Intelligence for Organic Chemistry, The DENDRAL Project. New York: McGraw-Hill (1980).
2. E.H. Shortliffe. Computer-based Medical Consultation: MYCIN. New York: American Elsevier (1976).
3. M. Minsky. A Framework for Representing Knowledge, in The Psychology of Computer Vision, edited by P.H. Winston, McGraw-Hill Book Company, New York (1975).
4. R. Fikes and T. Kehler. The Role of Frame-Based Representation in Reasoning, Communications of the ACM, Vol. 28, No. 9, 904-920 (1985).