EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN - PS DIVISION

EE

CERN/PS 93-52 (CO) /

See 9348

CERN LIBRARIES, GENEVA

CM-P00059546

9

NODAL - THE SECOND LIFE OF THE ACCELERATOR CONTROL LANGUAGE

G. Cuisinier, F. Perrriollat, P. Ribeiro
A. Kagarmanov, V. Kovaltsov *

* IHEP, Russia

International Conference on Accelerator and Large Experimental Physics Control
Systems
Berlin, Germany, October 18-22, 1993

Geneva, Switzerland
15/11/93

Nodal - The Second Life of the Accelerator Control Language

G. Cuisinier, F. Perriollat, P. Ribeiro (CERN, Switzerland)
A. Kagarmanov, V. Kovaltsov (IHEP, Russia)

Nodal is a popular interpreter language for accelerator controls since the beginning of the 70's. Nodal has been rewritten in the C language to be easily portable to the different computer platforms which are in use in today's accelerator controls. The paper describes the major features of this new version of Nodal, the major software packages which are available through this implementation, the platforms on which it is currently running, and some relevant performances. The paper presents the major domains of usage of Nodal and its capability for these classes of applications. The experience gained during the rejuvenation project of the CERN accelerator control systems is presented. The benefit of this use is discussed, in particular with a view to the prevailing strong constraints in personnel and money resources.

1. Introduction

Nodal was developed as an accelerator control language for the CERN SPS project. Since then it became a very popular tool for control, at CERN but also at other large laboratories like DESY or KEK. Within the project of rejuvenation of the PS and SPS control systems, it was decided to develop a portable version of the Nodal interpreter, written in C, which can be used on various platforms. This portable version must be able to reuse program and data files of the Norsk Data version. It also must handle the remote execution between old and new versions, and between the various platforms running under different operating systems.

This project was done in a large international collaboration involving CERN, Sincrotron Trieste (Italy), and IHEP Protvino (Russia). Some other laboratories took an important part in porting Nodal to various platforms. This paper will give a short overview of this portable version, and some results and experiences gained with this product.

2. Major goals and related options

The compatibility of text and data structures between various platforms with different data representations was addressed by the external representation. This representation is defined as a compact stream of double bytes, in the network order, using the IEEE floating point format. The first double byte, a "magic number", is used as flag to distinguish the ND 48 bit floating point format of the old version from the new one, and to do the needed conversion.

The external communication between the various platforms is provided via the TCP protocol. This protocol is accessed through a BSD socket interface which gives a uniform view on files and remote connections.

The portability is obtained through maintaining the structure of the initial version. The initial version was built around a small kernel which includes a limited number of basic commands. This kernel can be enhanced by add-on functions which can be seen as an extension of the command set. A standard list of functions make the basic Nodal. This structure is perfectly adequate for the portability goal, and the initial structure of the kernel and functions was reused in the C version. File system related facilities and network and operating system dependent features were well isolated in the source code files. The main features, sensitive to the OS, are keyboard input, multiplexing of input events, and process starting. The problem of the include file directory is well known for any multi OS package.

The modularity is given by a number of conditional compilation flags which select features to be included in the executable program.

Dynamic customization is an important issue. It is offered with a set of "program start option parameters", environment variables and configuration and profiling files.

On-line documentation is provided by dynamic extraction of information from a single source file which is the man page standard document. This is available only on UNIX platforms.

Performance is gained by intensive use of hash-tables for all stable lists of objects which are defined during the customization phase of the startup.

3. Features of the kernel

The kernel uses the double precision (8 bytes) floating point type for scalar variables, and all the basic numeric scalar C data types for arrays: byte, short (2 bytes), long (4 bytes), float (4 bytes floating point) and double float (8 bytes). Strings use the C style strings (NULL ended array of characters). Global data and shared Nodal functions are available on systems which offer a shared memory service. Extra relational operators have been added to provide numerical value comparison without approximation.

Asynchronous event processing capability is included in the system which provides multiplexing of input events. It can be used for dedicated services by a Nodal call back feature. This is a way to connect a Nodal block to events. This piece of code is executed asynchronously to the main flow of the program on arrival of an event. This feature is used to allow Nodal to do exception processing on system interrupt signals and on normal or abnormal program endings.

A minimum editing facility is provided according to the connected terminal capability. Full page editing facility is offered by an embedded call to the favorite editor of the user. A history list with navigation, save and restore capabilities, and a self learning and replay facility is available.

The command parser includes processing to easily build Nodal functions with variable parameter lists, (e. g. missing parameters replaced by defaults, variable number of parameters and variable parameter types: string, value or numerical expression). This facility is useful to build a convenient and uniform user interface to the equipment access package.

A pre-parsing feature was first introduced but then abandoned because the gain in performance was too marginal in comparison with what was gained with the hash coding technique.

The various flavors of the process (interactive, network servers, file driven, OS9 file server) are defined by option parameters in the start command. This command can also include a Nodal command which is executed at the end of the customization.

Intensive facilities to view the state of the various subsystems and internal conditions are provided for interactive use.

4. Optional services

A number of services can be included by conditional compilation flags according to the capability of the OS of the platform and the control system environment. Some of the services used in the CERN PS control system are shortly described.

Equipment access gives the capability to use all protocols available in the PS control software bus: command/ response in all variations with single and multiple data and equipment lists, asynchronous acquisition through subscription, access to reference values. A full access to an equipment real-time read-only data base is possible, with an easy way to scan working sets according to filtering conditions.

Weak process synchronization can be done by using the Nodal call back feature. Interprocess events and messages are available to communicate with other compiled or Nodal processes. This feature is built over X11 client messages.

Man machine interface is provided through X11 and the Motif tool kit facility widgets. Currently available to Nodal are: Main, all styles of Dialog box, Menu, Button, Scale, Drawing area, Label, Motif widgets and the CERN PS extension classes wheel-switch, Graph and Plot, and Digit. This facility includes Nodal call back on widget interaction and intensive access to widget resources. An easy way to draw an X, Y plot is available. It does not request any knowledge of the X11 facility, and it has an interactive customization facility and a code generator for better control of the display.

A very useful facility is the emulator of the CERN PS old (Norsk Data) operator console environment, intensively used for Nodal programs of the old control system. This facility is built on top of the X11 and Motif facility of Nodal and reduced considerably the porting effort for the renewal project. A similar emulator was built and used with the same success for the CERN SPS control conversion.

5. Status, performance and usage

The project of Nodal in C is completed. It has been ported on the main UNIX platforms for various flavors of UNIX: DEC Vax, Mips and Alpha (under OSF1), Apollo, HP SUN. It is also ported on the Vax VMS. For real-time platforms it is available for OS9 and LynxOS systems. It is also available under MS-DOS for 286 (and higher) PC platforms.

The average execution time for one Nodal line is 0.107ms on a DEC Alpha 3000/500 AXP, 0.236ms on a DEC Mips station 5000/125, and 1.375ms on a PC 386 25 MHz (under MS-DOS). However, only the relative values are meaningful. The performance for the DEC Mips 5000/125 is about 10 times better than the last ND120 used in the old PS control system.

The current usage of Nodal, beyond the very important one of old console emulation, is for hardware and software tests (for which it is a very powerful tool) and some quick prototyping. It is not

foreseen to write large control programs in Nodal as in the past.

7. Conclusions

Nodal is still a very useful tool for accelerator control, specially for testing and prototyping. Its easy integration into various control environments makes it attractive to preserve knowledge and skills of people who are not professional software developers.

The success of this project was widely due to the large international collaboration which is acknowledged here.

8. References

- "The Nodal system for the SPS", CERN 78-07, M.C. Crowley-Milling and G.C. Shering.
- "PS / SL Controls Consolidation Project", CERN Internal Report PS/91-09, CERN SL/91-12, PS and SL controls groups,
- "NODAL", CERN Internal Report PS/CO 91-20, G. Cuisinier, A. Gagnaire, F. Perriollat.