ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S1400P05

An Object-Oriented Implementation of the TRIUMF 92 MHz Booster Cavity Control System

N.A. Wilkinson and G.A. Ludgate
TRIUMF, 4004 Wesbrook Mall, Vancouver B.C., Canada, V6T 2A3

Abstract

A 92 MHz auxiliary accelerating cavity has been designed for installation inside the TRIUMF cyclotron, operating up to a maximum peak voltage of 200 kV. The cavity doubles the energy gain per turn for accelerating hydrogen ions in the energy region of 400-500 MeV, and reduces by 50% the stripping loss of the ion beam. The control system for the booster comprises a PC-based processor in a VME crate, for local control, and a 68030 processor with an ethernet connection as the interface to the TRIUMF Central Control System. The requirements for the booster control system were established by an object-oriented requirements analysis. Afterward, an object-oriented architectural design step was used to produce the processor allocation of the design, which was then implemented using C, for the VME processor, and a commercial database and screen generator product, for the VAX user interface.

I. INTRODUCTION

The RF Booster consists of an RF cavity, an RF amplifier, a transmission line connecting the amplifier to the cavity, a local control system and ancillary equipment. The RF cavity is made up of two symmetrical halves which are mounted on the lid and floor of the cyclotron vacuum tank and separated by 64 mm to provide a region free of all components. The accelerating voltage exerted by the RF cavity increases from 0 kV at a beam energy of 330 MeV to a maximum of 150 kV at a beam energy of 520 MeV. The RF cavity geometry is designed so that, when the booster is operating, each ion in the beam receives two impulses during its passage through the cavity. This energy gain enhances the turn separation of the beam, reduces the number of turns made during acceleration and reduces the beam loss due to electromagnetic stripping.

During the design and development of the booster, equipment specialists implemented a local control system based on an IBM-PC in VMEbus. Although this system met the requirements for local control of the booster by equipment specialists located in the RF area, its design did not consider the requirement for remote control by the cyclotron operators.

II. OBJECT-ORIENTED ANALYSIS

To identify the requirements for remote control of the booster, an object-oriented requirements analysis was carried out using the domain-driven specification techniques described in [1]. For this analysis, the notation of the Yourdon methodology with Ward-Mellor extensions[2] was used to represent objects, their behaviour and their interactions. Objects and their methods were represented by *data flow diagrams* (DFDs), object behaviour was represented by *state transition diagrams* (STDs) or *control transforms*, and information flows between objects were represented by *control* and *data* flows. A relevant aspect of the object-oriented methodology employed for the requirements analysis was the identification of objects which would interact with the RF Booster remote control system via information flows only. Such objects are known as *terminator* objects of the system.

As the only CASE tool available on-site was DECdesign, a CASE tool from Digital Equipment Corporation, this tool was used to capture the models of the booster remote control system. The strict Yourdon-Ward-Mellor methodology implemented by this tool did not easily accommodate the object-oriented paradigm chosen for the requirements analysis. As a consequence of this inflexibility, the desired goal of a purely object-oriented requirements document was contaminated by tool-specific considerations.

The DECdesign *verification* utility, however, proved invaluable in ensuring that control and data flows between objects were consistent and in ensuring that the data dictionary was maintained up to date.

III. THE RF BOOSTER REMOTE CONTROL SYSTEM

During the domain analysis it was recognised that personnel operating the booster would play two roles in their interaction with the booster: that of a *beam control* operator, whose only concern would be the affect of the booster on the beam, and the role of *equipment management operator*, whose concern was the management of the booster equipment.

These operator roles enabled the identification of the two domains spanned by the RF booster remote control system: the domain of beam control and the domain of equipment management. In the former domain, the requirements of the remote control of the RF booster by an *RF Booster Beam Control System* (RFB BCS) were examined and, in the latter domain, the requirements for remote management of the booster equipment by an *RF Booster Equipment Management System* (RFB EMS) were examined.

A. *The RFB Beam Control System*

Given the preceding assumptions, the purpose of the RFB BCS was described as follows:

- to enable operators in the Cyclotron Control Room to remotely control the phase and amplitude of the booster, and
- to provide operators in the Cyclotron Control Room with a meaningful display of the booster operating parameters necessary for remote control and monitoring of the booster.

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S1400P05

The diagram of Figure 1 illustrates the interactions between the RFB BCS and its terminator objects. Since, by mandate, only the RFB EMS was allowed access to the booster via the IBM-PC in VMEbus, the RFB EMS was identified as a terminator object of the RFB BCS. Within the RFB BCS, the object *RFB Booster Control* was created to represent the interaction of the RFB BCS with the RFB EMS terminator object. The only other terminator object identified was the RFB BCS Operator, and the object *RFB Operator Control* was created to represent the interaction of the RFB BCS with the RFB BCS Operator.

The state transition diagram of Figure 2 illustrates the

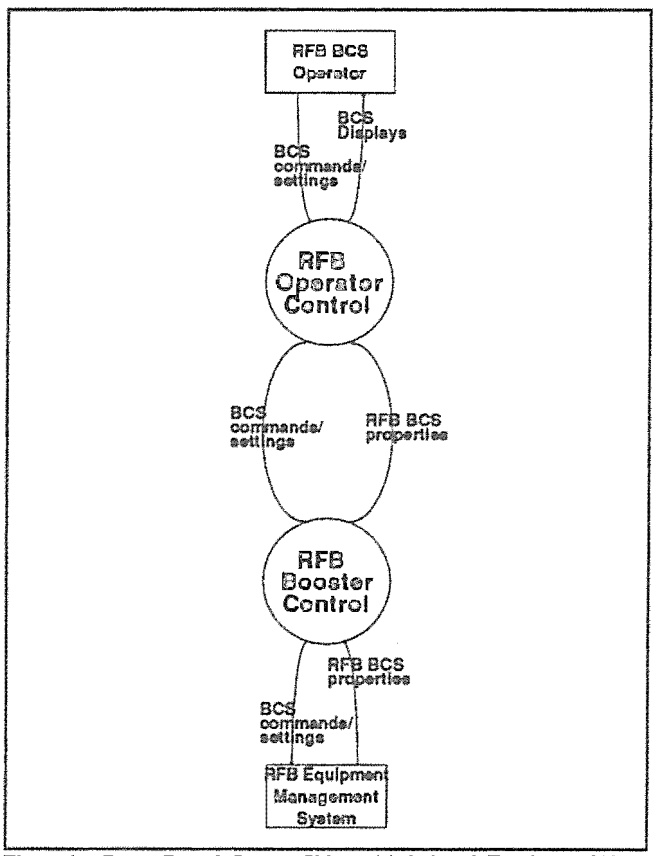

Figure 1: Beam Control System Objects (circles) and Terminator Objects (boxes)

model of the RFB BCS operator behaviour elicited from personnel interviews.

The requirements analysis for the RFB BCS has presently completed two external reviews and is considered ready for design and implementation.

B. The RFB Equipment Management System

An RFB EMS operator would, at times, require privileged access to the booster equipment and, by mandate, such access was limited to the IBM-PC in VMEbus. It was recognised, however, that the RFB equipment management operator would require remote non-privileged access to the booster equipment in order to manage other aspects of the booster equipment, such as alarms, warnings and data logging. Thus, the purpose of the RF Booster Equipment Management System (RFB EMS) was established as:

- to enable operators in the Cyclotron Control Room to remotely manage the booster and its equipment.
- to provide operators in the Cyclotron Control Room with a meaningful display of the operating parameters required for remote management of the booster and its equipment.
- to provide a data logging facility for the booster, and
- to serve as the functional interface between the RF Booster Beam Control System and the booster equipment.

The RFB EMS was found to interact with four terminator objects: the RFB BCS, the RFB EMS data log, the RFB EMS operator and the RF Booster. For each of these terminator objects, a corresponding RFB EMS object was created to represent the interaction between the RFB EMS and the terminator object. These objects and the essential information flows between them are shown on Figure 3.

The RFB EMS requirements analysis has presently completed one external review. During this review, it became clear that, whenever excessive beam spill is detected by the RFB EMS, it must interact with the cyclotron ion source and shut off the beam to prevent damage to the booster and ancillary equipment. The RFB EMS requirements are presently being revised to reflect this new requirement.

IV. Object-Oriented Design

In the design phase, one of the major steps in implementing an object-oriented architecture is the allocation of the requirements specification to technology units. Several implementation choices which affect this allocation have already been made, namely:

- the operator interface for both the RFB BCS and the RFB EMS will be implemented via a VAX/VMS commercial real-time database and screen generator product obtained from Vista Control Systems.
- the majority of the RFB EMS and RFB BCS objects will be implemented in a VAX/VMS system, as nearly all of the terminator objects with which these objects must interact are accessible there.
- communication between the VAX/VMS system and the VMEbus system will be implemented via the *socket server* ethernet protocol suite described in [3]. This suite implements message-based inter-process communication between VAX/VMS processes and processes residing in a VMEbus processor running the Unison real-time operating system from Multiprocessor Toolsmiths,
- since all of the code running on the IBM PC in VMEbus runs under MS-DOS, an ethernet equipped VMEbus processor card running the Unison real-time operating system will be added to the VMEbus crate to provide network communication between the VAX/VMS system and the IBM PC in the booster, and

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S1400P05

- the XDR (External Data Representation), which has been designated RFB1014 by the ARPA Network Information Centre, will be employed to define data formats for messages passed between the heterogeneous processors in the system.

In Figure 3, the aggregate information flow *RFB properties* consists of information which describes all properties of the booster. However, this information must be obtained from two different sources: the IBM PC in VMEbus and the TRIUMF Cyclotron Control System. The use of the message-based *socket server* network communication protocol allows information flows originating in the booster VMEbus system to be transparently encapsulated in messages between the VAX/VMS system and the booster VMEbus system.

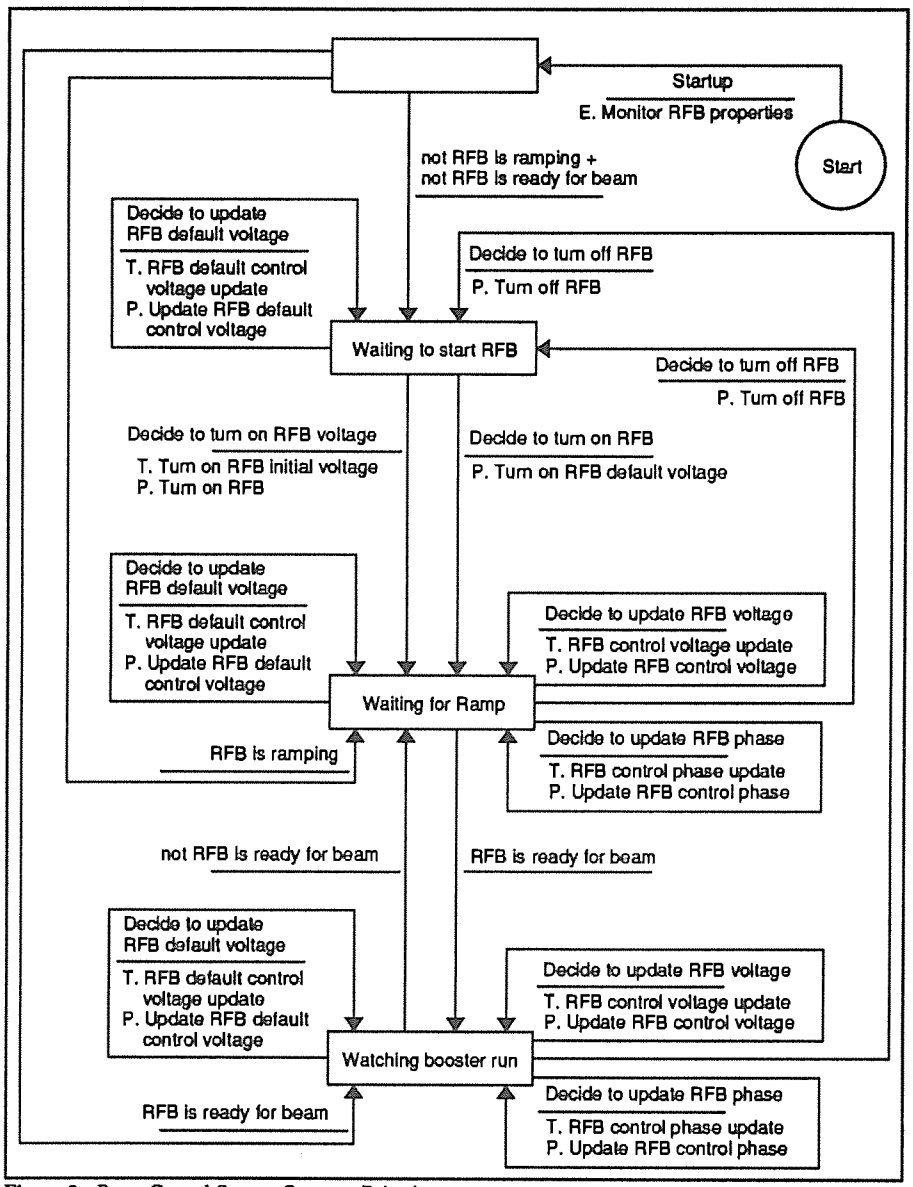

Figure 2: Beam Control System Operator Behaviour

V. OBJECT-ORIENTED IMPLEMENTATION

During the TRIUMF Beam Line 2C project, an approach was developed for the direct translation of object models into C code. The real-time data base from Vista Control Systems is central to the translation process as inter-object data and control flows, including flows from terminator objects, are implemented via channels in the database.

When implemented by this method, each *object* is composed of the following code and data:

- a *private data structure* describing the object instance. Included in this data structure is a record of the object instance *state*, a list of the database channels accessed by the object and a *state transition matrix* which describes the object's state transition diagram in tabular form.
- a VMS Asynchronous System Trap (AST) handler, which serves as a *transaction centre* for object *events* and/or *conditions*, and
- the object *methods* which implement the object's behaviour as C function calls.

An object attaches its AST handler to each database channel from which it expects to receive a data or control flow, and the AST *user parameter* identifies the event which is associated with the channel. (In the case of condition or data flows, this event signifies a *change* in the condition or data flow.) An STD is directly translated: a transition condition on an STD becomes one or more events associated with database channels; the object instance state is maintained in private memory; and a transition action on an STD becomes the object method associated with a *state-event* pair in the state transition matrix. When an event occurs, the object's AST handler serves as a transaction centre by using the event and the object instance state to look up the *destination state* and associated method in the state transition matrix. The associated method is triggered by the AST handler and the destination state is recorded as a

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S1400P05

new value of object instance state. Once the function call to the associated method is complete, the AST handler checks for transition conditions which cause a transition to a new state. If a transition condition which causes a state transition is satisfied, the AST handler updates the object instance state and continues checking for conditions causing state transitions until none are found. When no valid transition conditions are found, the AST handler exits.

The considerable overheads inherent in VAX/VMS process creation precluded creation of separate processes for object methods which were continuously active. Instead, continuous activities were emulated by calling any enabled activities during execution of the AST handler.

The use of real-time database channels for information flows between objects enables concurrent and independent code development and testing. For instance, during the TRIUMF Beam Line 2C project, all information flows between objects were defined as database channels prior to the generation of any code. The coding and testing of objects did not depend on the existence of other objects because each object referred to appropriate database channels for its information flows, rather than to the objects from which the information flows originated. This enabled concurrent code development and testing on an object-by-object basis.

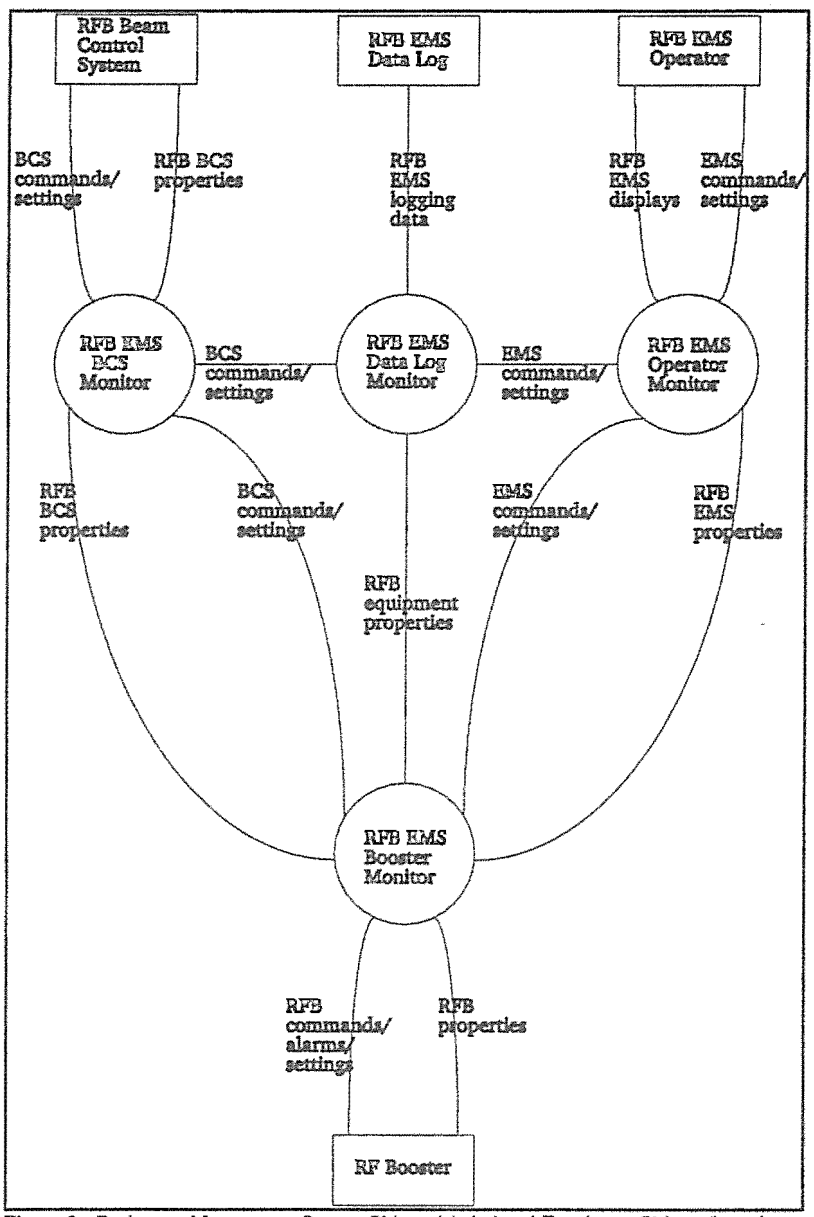

Figure 3: Equipment Management System Objects (circles) and Terminator Objects (boxes)

VI. CONCLUSIONS

The object-oriented analysis of the RF booster remote control system has produced a requirements specification which can be directly translated into code. However, the use of an inappropriate CASE tool (DECdesign) caused serious tool-specific distortions to enter into the specification for the booster remote control system.

The use of a message-based network protocol has enabled the transparent encapsulation of information flows which are obtained via the network. This has considerably simplified the design architecture of the RFB EMS by eliminating the need to distribute objects between processors.

The implementation of RFB EMS and BCS objects by direct translation of the requirements specification is presently being carried out. Object implementation is proceeding concurrently and independently, since channels in the real-time database used for information flows between objects can be defined before any objects are coded.

References

[1] C. Inwood, G.A. Ludgate, D.A. Dohan, E.A. Osberg and S. Koscielniak, "Domain-Driven Specification Techniques Simplify the Analysis of Requirements for the KAON Factory Central Control System," *Nuclear Instruments and Methods in Physics Research*, Vol. A293(1,2), Amsterdam: Elsevier Science Publishers, 1990, pp. 390-393.

[2] P. Ward and S. Mellor. *Structured Development for Real-Time Systems*, Yourdon Press, New York, 1985.

[3] N. Wilkinson and D. Dohan, "Synchronous Message-Based Communication for Distributed Heterogeneous Systems," submitted to: *International Conference on Accelerator and Large Experimental Physics Control Systems*, November 11-15, 1991, Tsuuba-shi, Ibaraki-ken, Japan.