APPLICATION OF OBJECT ORIENTED TECHNIQUES IN THE TRIUMF BEAM LINE 2C CONTROL SYSTEM

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

ABSTRACT

The KAON Factory central control system study employed a uniform approach to requirements analysis, architectural design and programming based on well established object oriented principles. These principles were applied to the successful analysis, design and implementation of the control system for the TRIUMF Beam Line 2C Solid Target Facility. The specification for this control system was created in collaboration with Beam Line 2C equipment management experts and, once the analysis models were validated, an approach was developed for the direct translation of these models into C code. A commercial real time database was central to this translation, as inter-object data and control flows are implemented by channels in the database. This paper focuses on the experience gained in the use of object oriented techniques during the complete analysis–design–implementation cycle of a working control system and on the utility of implementing such a system using a commercial real time database and graphical interface.

1. INTRODUCTION

Beam Line 2C[1] uses a proton beam from the TRIUMF Cyclotron to produce radioisotopes. The Beam Line 2C *Solid Target Facility* (BL2C STF) irradiates solid targets and consists of a water system, a beam energy degrader and various pipes, pumps, valves, sensors and actuators. The water system is a significant component of the STF and is composed of an *elevating* water subsystem, a *cooling* water subsystem, a nine metre water column and associated hardware. The elevating water subsystem is used to insert and remove the solid target, and the cooling water subsystem is used to provide shielding and cooling while the solid target is being irradiated.

When a solid target is to be irradiated, it is mounted on a float and water from the elevating system reservoir is pumped into the water column until it is full. The assembled target and buoyant float are placed in the water column and the elevating water is drained from the water column. As the water drains from the water column, the target/float assembly descends with the water until it reaches the irradiation level, where it is latched in position. Once the column is empty of elevating water, water from the cooling water system is pumped in and continually circulated through the water column while the target is being irradiated.

The water column accepts water from both water subsystems. However, mixing of cooling and elevating water in the water column must be avoided because the cooling water may be radioactive and the volume of water in the two water subsystems exceeds the reservoir capacity of each subsystem. A spill of radioactive cooling water may result if water originating in the reservoir of one subsystem is inadvertently drained from the water column into the reservoir of the other subsystem. One of the responsibilities of the STF control system is to ensure that this does not happen.

2. OBJECT ORIENTED ANALYSIS

The Beam Line 2C control system requirements were specified via an object oriented methodology which was developed during the KAON Factory Project Definition Study.[2)3)] This object oriented methodology uses the notation of the Yourdon methodology with Ward-Mellor extensions[4)] to describe objects according to their static characteristics, their dynamic characteristics and their mutual interactions.

The static characteristics of an object can be identified when the object is not changing as a consequence of interactions with its environment.* These characteristics include the object's name, its attributes, its relationships with other objects and its potential relationships with other objects. Since the extended entity relationship diagram[5)] (EERD) graphically represents entities, their attributes, their relationships and their potential relationships, the EERD is suitable for modelling the static characteristics of objects.

In contrast to static characteristics, the dynamic characteristics of an object can be identified by observing an object's response to stimulation from its environment. Since *behaviour* is defined as *anything an [object] does involving action and response to stimulation*,[6)] it is clear that an object's behaviour is key to any description of its dynamic characteristics. The dynamic response to an external stimulus is *mode* dependent. For instance, a person who is unconscious will respond to a different set of stimuli than the same person would when conscious. The state transition diagram[7)] (STD) can be used to represent this mode dependent behaviour. When an external stimulus occurs, the STD indicates whether or not the object will respond to the stimulus. The STD also indicates *how* the object will respond by identifying what actions will be done and/or what activities will be started or stopped.

A complete description of an object is obtained using the object structure diagram (OSD), which is shown in Figure 1. On the OSD, object behaviour is represented by a behaviour transform, object actions and ac-

*Present address: Superconducting Super Collider Laboratory, 2550 Beckleymeade Ave., Dallas, TX, USA 75237

*i.e. when the object is in *equilibrium*

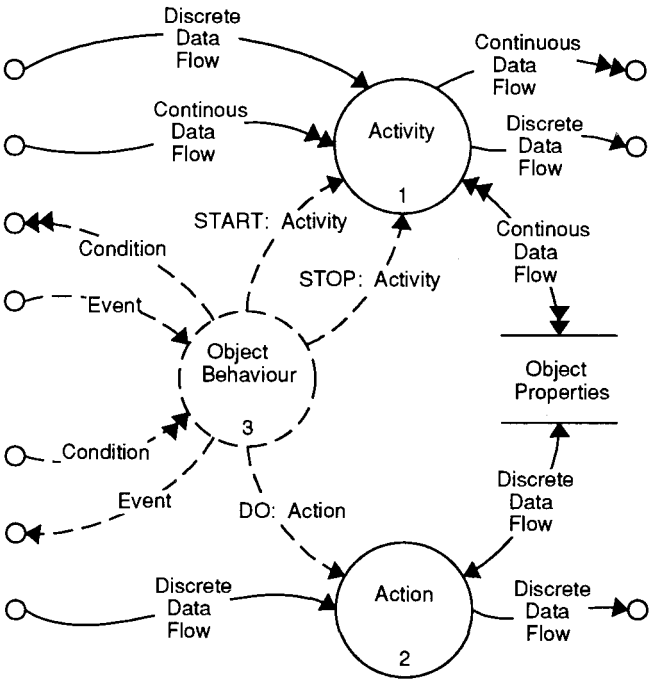

Figure 1. **Sample Object Structure Diagram**

tivities are represented by data transforms, the object's static properties are represented by a data store and information flows are represented by directed arrows. The behaviour transform encapsulates a state transition diagram, and the data store encapsulates the object EERD. OSD information flows are classified according to their persistence over time and their information content. Information flows which persist over time are represented with double arrowheads; information flows which do not persist over time are represented with single arrowheads. An information flow which contains data is classified as a data flow and is represented with a solid line. Events, conditions and activation prompts are represented by dotted lines. Conditions and continuous data flows persist over time. Discrete data flows, events and activation prompts do not persist over time. The small bubbles on Figure 1 indicate information flows which cross the object's boundaries. Whereas data flows, events and conditions may cross object boundaries, activation prompts are always confined within objects. (Activation prompts start or stop object activities and/or activate object actions.)

Object interactions are modelled on an object interaction diagram (OID). On the OID, the representation of inter-object information flows is the same as on the OSD. However, it is on the OID that objects which transform mass and energy (*terminator* objects) are distinguished from objects that transform information (*system* objects). Each terminator object on an OID is represented by a box and each system object is represented by a data transform†. Each terminator object is paired with a system object, and this system object mediates between the terminator object and other system objects.

3. BL2C CONTROLS REQUIREMENTS

The uniform, object oriented methodology described in Section 2 was applied to the requirements for the BL2C STF Control System. As the only CASE tool available on site was DECdesign, a CASE tool from Digital Equipment Corporation, this tool was used to prepare the BL2C STF models. The strict Yourdon–Ward–Mellor methodology implemented by this tool did not accommodate some aspects of the object oriented methodology. For instance, the OID should show both system and terminator objects, but DECdesign only allows terminator objects to appear on the Context Diagram. As a consequence of this inadequacy, only EERDs and STDs were used to describe the BL2C STF objects. The OID and the object OSDs were inferred from the EERDs and STDs.

To determine the BL2C STF controls requirements, BL2C equipment management experts were interviewed in detail. Their responses were analysed to identify BL2C STF objects, their static characteristics and their dynamic characteristics. EERDs and STDs were used to describe these characteristics and were subject to ongoing review and verification by the equipment management personnel.

During object model verification, additional objects were identified. The initial analysis of the BL2C STF produced an EERD which showed only a single, monolithic water system. This view did not adequately describe the two water subsystems and the water column. As the requirements analysis progressed, it was evident that the water system model should reflect this structure. In retrospect, more effort should have been devoted to the analysis of the static characteristics of the STF.

4. THE LOW LEVEL BL2C CONTROLS

The BL2C hardware is accessed via serial and parallel CAMAC systems. Concurrent with the analysis process, a low level interface to the BL2C STF system was created using a VAX/VMS commercial real time database and screen generator obtained from Vista Control Systems (VCS). This interface enabled the BL2C equipment management experts to run the system by direct manipulation of the CAMAC hardware. The VCS real time database is made up of a number of *channels*. Each channel may represent an analogue, binary, or integer value or an array of values. Periodically‡, a program updates the database and the corresponding hardware to ensure that they are consistent with each other. VCS provides an X windows screen generator to allow development of graphical interfaces to database channels. VCS also provides libraries of system calls which can be made from user software (in C or Fortran) for access to

†which explodes to the object's OSD

‡typically at 1 second intervals

database channels. One of the many facilities offered by the VCS system calls is the ability for a user supplied VMS Asynchronous System Trap (AST) handler to be *attached* to a database channel. Once an AST handler is attached to a channel, it will be called whenever the value of the channel changes. Upon being called, the AST handler may be passed a parameter, which is defined by the user when the AST handler is attached to the channel.

5. IMPLEMENTING BL2C OBJECTS

Once the object models were validated, inter-object information flows, including flows from terminator objects, were defined as channels in the real time database. Control flows and conditions were implemented as binary channels. Discrete data flows and continuous data flows were implemented as analogue channels. Inter-object communication is implemented by attaching an AST handler from each object to the relevant channels in the real time database. The AST parameter mentioned in Section 4 is used to identify the corresponding information flow. This mechanism ensures that each object is informed of the arrival of events and discrete data flows or of the change in a condition or a continuous data flow.

STF object behaviour is implemented by direct translation of the object STD into a state–event–action table, which is described by a C structure. The object instance state is maintained in private memory and a transition condition on an STD becomes one or more events associated with database channels. The transition actions associated with each state–event pair are implemented by a C function. Using this translation schema, each BL2C STF object is composed of:

- 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 their values, and a state–event–action table.
- the VMS Asynchronous System Trap (AST) handler, which serves as a transaction centre for object events and/or conditions, and
- the C functions which implement the object's actions and activities.

When an AST occurs, the associated event is indicated by the AST parameter. The event and the current object instance state serve as an index into the state–event–action table. The AST handler uses this index to look up the new state and the address of the C function which implements the actions associated with the state–event pair. The new state is recorded as the current object instance state and the associated C function is called. The AST handler then checks for transition conditions which may now be valid. If a valid transition condition is found, the AST handler again looks up and calls the C function associated with the transition and updates the object instance state. When no more transition conditions are satisfied, the AST handler exits.

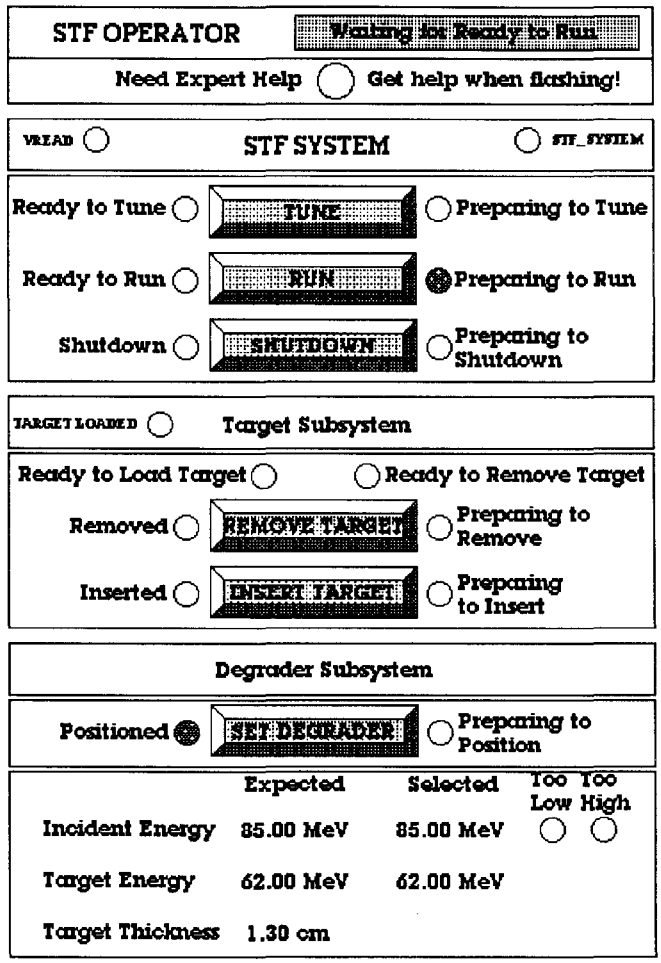

Figure 2. **STF Operator Control Panel**

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

6. DISCUSSION

A significant aspect of the analysis was the modelling of the BL2C Solid Target Facility Operator. Modelling human behaviour is challenging because of the large number of modes involved and because of the wide range of behaviour which, typically, must be modelled. However, if the specialist knowledge about how to operate a system can be captured and accurately modelled in an STD, the STD can be implemented. By capturing this information and implementing the operator *as a system object*, the complex sequences and interactions which would require an expert operator can be considerably simplified to the point that a non-expert can safely operate the system. In the case of the BL2C STF opera-

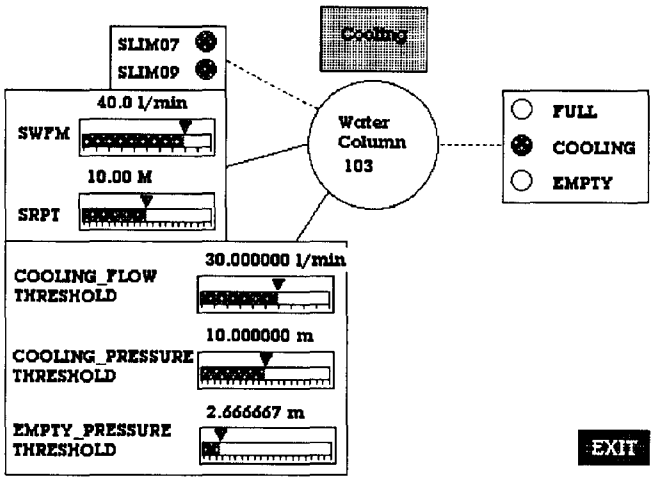

Figure 3. **Water Column Object Test Page**

tor model, the interactions required to operate the BL2C Solid Target Facility were reduced to only six pushbuttons and three numerical inputs, as shown on Figure 2.

The STF operator *object* can recognise dangerous or unsafe situations and, by popping up a dialogue box, inform the operator that an equipment management expert is required. Other dialogue boxes are used to inform the operator of the necessity to manually load or remove a target/float assembly.

The use of real time database channels for information flows between objects enabled concurrent and independent code development and testing during the BL2C implementation phase. Since all inter-object information flows are defined as database channels, each object refers to appropriate database channels for its information flows, rather than to the objects from which the information flows originate. Thus, the coding and testing of objects did not depend on the existence of other objects and could proceed concurrently on an object-by-object basis.

Figure 3 illustrates a test page used for verification of object behaviour during code development. This test page represents a fragment of the BL2C OID. By convention, all input information flows to the water column object are shown on the left hand side of the test page and all output flows are shown on the right hand side. The water column object is represented by the large circle on Figure 3 and the water column state (*Cooling*) is shown in the box above this circle. The small circles labelled *SLIM07* and *SLIM09* are pushbuttons controlling binary channels. These channels can be toggled by clicking the mouse on the corresponding circle. In this manner, events and condition changes may be simulated. The remaining inputs to the object are analogue channels which represent data flows. These may be changed by mouse and/or keyboard input, again simulating an input to the object. The water column object behaviour can be *exercised* by simulating the various combinations of input events, conditions and data flows. As the inputs change, the state changes and output information flows can be observed and verified by direct comparison with the object's STD. This technique facilitated the thorough testing of each object in isolation, rather like component testing of an electronic assembly. Only when an object passed its component test was it installed in the system.

7. CONCLUSIONS

The object oriented analysis of the BL2C STF control system produced a requirements specification which could be directly translated into code. In particular, the information captured by modelling operator behaviour allowed implementation of the complex operator control sequences required to operate the facility as part of the operator *object*. This enables safe operation of the system by non-experts.

The use of the real time database from Vista Control Systems to implement inter-object information flows allowed concurrent and independent implementation of the BL2C STF control system objects. This implementation would have taken a single programmer nine months. Instead, three programmers were able to use the object oriented requirements specification to concurrently implement the STF control system objects in only three months.

8. REFERENCES

1) D R Pearce. "A Facility for Radioisotope Production at TRIUMF with 70/110 Mev Protons". Submitted to: *XIII International Conference on Cyclotrons and their Applications, Vancouver, British Columbia, Canada, 6-10 July 1992.*
2) D A Dohan, G A Ludgate, E A Osberg, S Koscielniak, and C Inwood. "Definition Study of the TRIUMF KAON Factory Control System Project". *Nuclear Instruments and Methods in Physics Research*, A293(1,2):6–11, 1990.
3) 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*, A293(1,2):390–393, 1990.
4) Paul T. Ward and Stephen J. Mellor. *Structured Development for Real-Time Systems*, volume 1. Yourdon Press, 1985.
5) E A Osberg, G A Ludgate, S Koscielniak, and D A Dohan. "Dynamic Object Modelling as Applied to the KAON Control System". *Nuclear Instruments and Methods in Physics Research*, A293(1,2):394–401, 1990.
6) Frederick C. Mish, editor. *Webster's Ninth New Collegiate Dictionary.* Merriam-Webster, Springfield, Mass, USA, 1988.
7) Paul T. Ward and Stephen J. Mellor. *Structured Development for Real-Time Systems*, volume 1, chapter 7, pages 64–80. Yourdon Press, 1985.