A RULE-BASED CONSULTANT FOR ACCELERATOR BEAM SCHEDULING USED IN THE C.E.R.N. PS COMPLEX

J. Lewis, P. Skarek, L. Varga[1]
PS Division, CERN, CH-1211Geneva 23, Switzerland

ABSTRACT

The CERN PS accelerator complex consists of nine interacting accelerators which work together to produce particle beams for different end users, varying in particle type, energy, time structure, and geometry. The beam production schedule is time sliced and depends on the current operational requirements and dynamically on the accelerator status, so that production schedule changes occur in real time. Many potential schedules are not valid due to various system constraints and these constraints vary over time as new operational modes are introduced. In order to ensure that only valid schedules are given to the complex, an automated tool has been developed to indicate whether a potential schedule is valid or not. This presentation describes the method by which the validity of a beam schedule is determined and how this method was implemented using a rule-based approach based on SQL, avoiding the use of an expert system shell. Both the data to instantiate the rules and the rules themselves are kept in an Oracle data base. The SQL interpreter provides the inference engine for this knowledge-based system. A few examples are presented and the running experience with the tool is discussed.

1. INTRODUCTION

The CERN PS complex is a network of nine accelerators, which has been described as a factory for producing particle beams [1], [2]. Each accelerator is scheduled in 1.2s time slices called basic periods and works on the same beam for one or more of these periods, which constitutes an accelerator cycle. Every cycle requires a characteristic setting of the accelerator parameters (Pulse to Pulse Modulation [1]), which is chosen according to a telegram message sent to each accelerator at the beginning of the cycle. The beams produced by the PS accelerator network may thus be considered as lists of cycles to be executed sequentially on a set of its accelerators. These beams can be replaced in real time by an alternative beam set, depending on a set of external conditions representing each accelerators hardware status, and the current operational requests. The operational requirements for the complex are input to an editor via a Graphical User Interface as a time-ordered set of beams and alternatives called the Beam Coordination Diagram (BCD) [2]. The BCD is then executed by the Master Timing Generator according to the external conditions.

2. BEAM SCHEDULES

Although many BCDs can be built with the editor, not all of them can be executed correctly by the complex because the physical properties of the accelerator equipment pose certain constraints. Some constraints are concerned with individual cycles in isolation, while others are more complex and are concerned with cycle sequences called supercycles. The BCD Editor enforces some simple constraints on cycle and supercycle composition, but since this is not enough to guarantee an executable BCD, a second level of checking was required, Fig-1. For example:

- *A PS user LEAR (Low Energy Antiproton Ring) containing the PBAR (Antiproton) option should follow a cycle longer than 1.2s*

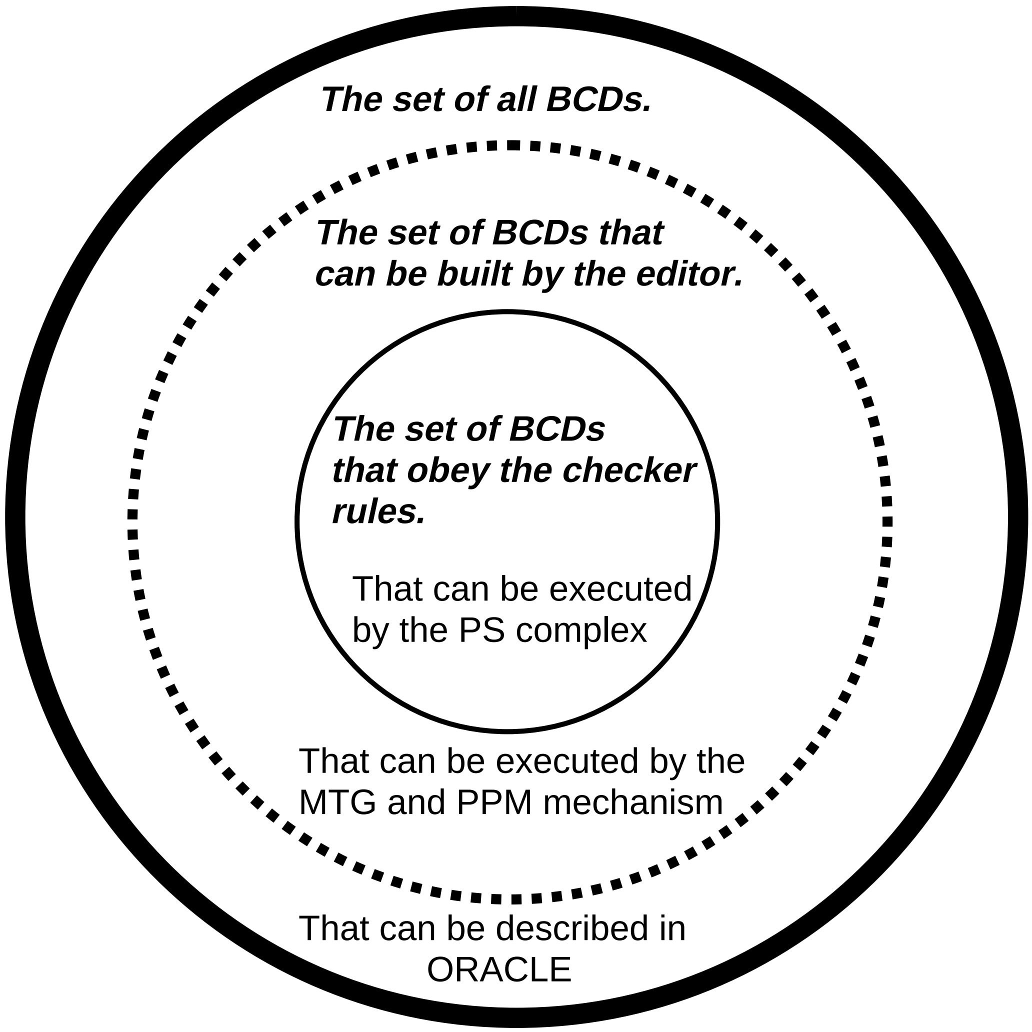
  A constraint concerning the minimum ramp time for some power supplies involved in transferring the beam and hence a restriction on cycle sequence.

[1] On leave from K.F.K.I.-M.S.Z.K.I. , Budapest, Hungary.

- *Two PS users AA (Antiproton Accumulator) should be separated by a time difference which is a multiple of 4.8s*

  A constraint concerning average power consumption and hence magnet temperature, giving a cycle sequence restriction.

Figure 0-1 Different levels of constraints for the BCD



3. A RULE BASED APPROACH

The requirements discussed above could not be satisfied easily by a traditional program, so we took a rule-based approach, but we did not want to have to maintain a full expert system shell and fortunately the constraints do not need rule chaining, because the rules are independent. The BCDs and the rules defining the constraints are stored in an Oracle database and we use the SQL language to check that the BCDs satisfy these constraints. However the direct use of SQL is not recommended for an operator and some concepts, such as the time delay from one cycle to another require complex SQL expressions. We developed a rule language which is similar to SQL and has special expressions for the BCD concepts. This approach led to the development of two programs: the BCD rule checker and the BCD rule editor. The rule editor allows the user to create rules using the formal rule language and compiles them into SQL expressions to be used later by the checker. The SQL interpreter plays the same role that an inference engine plays in an expert system shell and the knowledge base, the other essential part of an expert system, is the set of rules stored in Oracle together with the properties of the BCD used in the rules to instantiate the rule variables.

The constraints on valid BCDs can be expressed by rules which have four text fields: the informal text, the formal text, the warning text, and the SQL text. [Fig-2] The SQL text is normally not editable, it is automatically generated by the BCD Rule Editor program and the BCD Rule Editor normally does not show this part of the rule to the user. The BCD Checker program uses this SQL text to check the BCD. The other three text fields can be edited using the BCD Rule Editor. The informal text and the warning text parts have no syntax restrictions. The warning text usually describes in one sentence the problem and is displayed with the results and the informal text describes and comments on the rule. The formal text field contains the rule itself and must satisfy a syntax which is checked when the rule is saved and the BCD rule editor generates the SQL text from it.

Figure 0-2 An example of a Rule

Rule no: 2
Date: 3rd Aug 1995 10:30 AM
Enabled: Yes
Informal text: A PS user LEAR contains a harmonic different from 10
Formal text: cycle set A exists where ('CPS' = A.machine and 'LEAR' = A.username and not 'H10' = A.plsgroup.HARMN) ;
Warning text: Wrong harmonic number for this user
SQL text: select A.cyclemachine, A.type, A.cyclestart from tmp1 A where 'CPS' = A.cyclemachine and 'LEAR' = A.username and not 'H10' = (select t.plsgroupvalue from temp2 t where t.bpkey = A.bpkey and t.plsgroup = 'HARMN')

The formal text imposes conditions on a 'cycle set' and if there is a set of cycles that does not satisfy certain conditions then a warning has to be given. In the above example the name of the cycle set is 'A' and the cycles of this set can be referenced later in the condition part using this name. The condition is given after the 'where' symbol and it is a Boolean expression. Cycle properties used in equations are formed by concatenating the name of the cycle and the name of the property with a dot. For example A.machine is the machine in which the cycle is executed or A.username is the user of the cycle. An example of the value that the A.username property can take is: 'LEAR'.

4. CONCLUSION

We have developed and implemented a technique for checking the validity of a beam schedule. This technique gives the flexibility of a rule-based system, but does not have the overhead of an expert system shell. The implementation includes a rule description language, a user-friendly rule maintenance and editor program and an inference program that applies the rules and presents the results in graphical form.

The first few months experience has shown that the users can create BCDs with a higher reliability. The rule-based approach has been proven to be sound and gives enough flexibility to express most constraints on BCDs. New rules can easily be created by the users. The speed of the inference program (in fact Oracle access and SQL interpretation) is a limitation, even though there are no real time requirements. Steps will have to be taken in the future to combine all the rules into a single optimized SQL query. For the future we consider it important that this diagnostic tool become part of the general environment for timing diagnosis and recovery actions [Ref. 3].

ACKNOWLEDGMENTS

We acknowledge the help of I. Campos during her stay at CERN on leave from DFTUZ University of Zaragoza, Spain, in helping with coding and documentation.

REFERENCES

[1] F. Perriollat, C. Serre: "The new CERN PS control system overview and status", ICALEPCS, Berlin, Germany, October 18-23, 1993, Nucl. Instr. And Meth. A352 (1994) 86.

[2] J. Lewis, V. Sikolenko: "The new CERN PS timing system", ICALEPCS, Berlin, Germany, October 18-23, 1993, Nucl. Instr. And Meth. A352 (1994) 91.

[3] P. Skarek, L Varga: "Multi-Agent Cooperation for Particle Accelerator Control", Accepted for presentation on the 3rd World Congress on Expert Systems, Seoul, Feb 5-9 1996.