The Fermilab Sequencer Use in Collider Operations

Jerry Annala
Fermilab
Main Accelerator Department
MS 341
P.O. Box 500
Batavia, IL. 60510 USA

ABSTRACT
At the Fermilab collider an elaborate setup procedure precedes each colliding beam store. The entire operations crew works for about 2 hours between stores insuring that each accelerator is properly prepared for the collider fill. The process relies on a great deal of high level software. The most consequential program used during this time is the Colliding Beams Sequencer. To understand the impact the sequencer has on colliding beam operation one must understand the basics of how the collider runs, its history and even a fair amount of detail about the sequencer itself.

DEFINITION OF SEQUENCING
Sequencing refers to managing a set of complex tasks that are to be completed in sequential order. This refers to a very high level set of tasks, not fundamental tasks carried out on the microprocessor level. An example of such a task is to run a dedicated application program to adjust devices to minimize steering errors in an injection line.

SHORT HISTORY
In the early days of the collider, the person responsible for developing a way to control the accelerators in collider mode visited CERN and returned with ideas about how the application programs should look that loaded the waveform generators in the field[1]. The programs should convert the physics parameters such as "tune" into waveforms that power supplies should follow[2]. This person also wrote an early version of the sequencer to intelligently communicate with these programs that loaded waveforms for the power supplies.

To prepare the accelerators for a store the operators would run the sequencer program and follow a written check list. Each operator would have his/her own check list that would be between 2 and 8 pages long. This checklist was designed to make sure that all necessary steps were completed in the right order. Most of the checklist was filled with very mundane activity such as starting plots of important parameters and turning on or off individual devices. If the set up procedure ever changed, the Operations Department Head would edit the check list and then insure that the operators received only the most recent version. An operator who was most familiar with the setup procedure would still have to follow the checklist very carefully to insure that no procedural changes were missed.

In 1990 the collider was upgraded to run with Protons and Antiprotons on separated orbits. With this upgrade came an upgrade in the control system as well. There was new controls hardware and new Sequencer to make the collider fill process more robust. The design of the upgraded procedure was based on the following ideas or philosophy.

1) Operation should be deterministic. Operators should make decisions only at predetermined times and the rest of the collider setup is completed ahead of time, "off-line".

2) The real time control should be pushed as far as possible into the embedded systems in the field.

3) The operation should be mostly single threaded. All of the accelerators are prepared for a transfer in parallel, but each machine is prepared in a fairly rigid sequence.

4) The sequencer should be able to coordinate all of the complex tasks involved with filling the collider as well as executing all of the simple mechanical activities.

The author of the new sequencer was a former operator who knew what type of program worked in the control room and was not willing to accept awkward ways of doing things such as those that are developed in a commissioning environment. He worked with the eventual users of the program to better define its look and function. An indication of the success of the controls upgrade is that the paper check list disappeared, and its function is completely contained in the new sequencer. The philosophy spelled out above was not new with the upgrade, but the upgrade set out to better allow for these ideas to be implemented.

A key part of the controls upgrade hardware was a new type of waveform generator known as a CAMAC 465 series module. This card contained tables that would play as a function of time or machine data (such as beam energy). The card accommodates 32 interrupt levels. Each level could contain an equation containing a time table and up to two equations containing machine data tables. All three equations associated with a particular interrupt level would play in unison and add linearly. Each table can accommodate 64 break points. These cards are deep enough to allow the entire collider fill process to be programmed ahead of time. A clock system that broadcasts events to the entire accelerator then triggers these interrupt levels. This type of waveform generator allowed moving the real time control to the embedded systems in the field, as well as predetermining the collider operation.

The COLLIDING BEAMS SEQUENCER

The colliding beams sequencer is the program that orchestrates the execution of the preloaded tables. The sequencer is built to enforce the single threaded design of the collider. The program user interface is shown in figure 1 There is a list of "aggregate commands" in the left had column. These represent major blocks of activities performed during the setup for a fill. The right hand column shows the individual commands that make up an aggregate command.

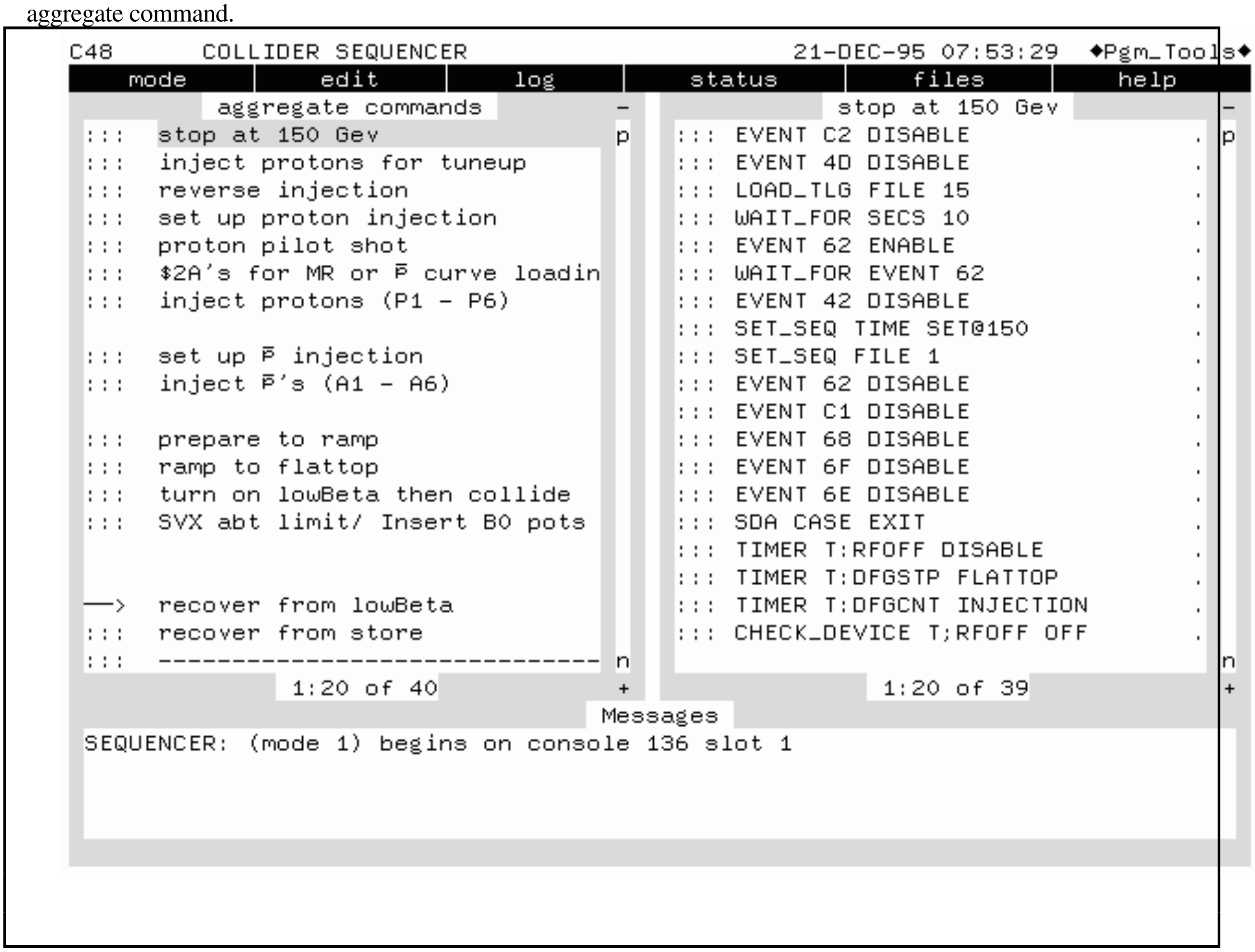

FIGURE 1 Sequencer Program Image

An aggregate command can be executed by clicking on the dots to the left of the aggregate. The individual commands begin executing in order from the top. Each command will execute unless an error is encountered within one of them. If an error occurs, the sequence halts so the operator can resolve the error. The sequence then can be continued from the place it halted by clicking on the three dots to the left of the individual command.

When an aggregate command finishes, an arrow moves to the next logical aggregate to execute. The movement of the arrow controls the normal flow of the collider setup process. The actual sequence of the aggregates that is guided by the movement of this arrow is determined by the machine physicist who builds the commands. The commands are built using the sequencer by clicking on the "edit" menu item. On-line help and a context sensitive editor guides the user through the building of commands.

Some of the individual commands are actually smaller ones packaged together by the programmer. An example of one of these complex commands is "inject". Figure 2 shows the on-line help for the inject command. This type of help is available for all allowed sequencer commands. This command is made up of several more fundamental ones. Since we include the inject command 19 times in the sequencer, consolidating this list of individual commands into a complex one greatly reduces the number of commands needed, and makes the function of the sequence more clear to the user. Also, inside of complex commands the programmer provides exception code. If the inject command incurs an error during execution, all functions in it will execute before the error message is posted and the sequence halted. If the commands were entered individually into the aggregate, the initial error would halt the sequence immediately leaving the Tevatron in a vulnerable state.

```
                         INJECT command help

INJECT sends a bunch into the TEVATRON from the MainRing or Accumulator.
Special actions indicated by:
  + -> error for this command does not stop execution of the list
  # -> this command is presently bypassed--it will not be executed

PROTON:+SET_DEVICE T:SBDEVT -2          set SBD event
       #SET_DEVICE T:SBDBNM <bunch>     set SBD bunch
        WAIT_FOR SCMOD <delay>          wait for delay modulo 40 into cycle
        COG PROTONS <bunch>             cog proton bunch
       +SDA CASE <bunch>                arm proper SDA case
        EVENT 4D ENABLE                 enable event $4D
        WAIT_FOR EVENT 5C               wait until event $5C is seen
        EVENT 4D DISABLE                disable event $4D
       +CH13_MESSAGE                    post a ch13 message
        ($4D is also disabled if command fails part way through its execution.)

PBAR:   Turn M:RPGON ON or OFF depending on the reading of A:IBEAM.
        SDA CASE (bunch)                arm proper SDA case
       +SET_DEVICE T:SBDEVT -1          set SBD event
       #SET_DEVICE T:SBDBNM <bunch+6    set SBD bunch
        SET_DEVICE A:R264F1 <delay>     set center frequency for Accum. RF
        WAIT_FOR SCMOD 35 % 40          wait for 35 secs before 40 sec marker
        EVENT 9A TRIGGER                trigger $9A (if no event-protection on)
        COG P-BARS <bunch>              cog pbars bunch
        EVENT 40 ENABLE                 enable event $40
       +CH13_MESSAGE                    post a ch13 message
        WAIT_FOR EVENT 2A               wait for p-bars to extract
        WAIT_FOR SECS 1.0               pause
       +CUSTOM COOL_GAIN                set all cooling gains
       +SET_DEVICE A:CMPA01 -= 0        reduce A:CMPA01 setting by 2
       +SET_DEVICE A:CPPA01 += 0        increase A:CPPA01 setting by 10
        WAIT_FOR EVENT 5B               wait until event $5B is seen
        EVENT 40 DISABLE                disable event $40
        TIMER T:SAPPBS DISABLE          disable timer
       +CH13_MESSAGE                    post a ch13 message
        ($40 is also disabled if command fails part way through its execution.)

<particle>      PROTONS/PBARS   inject protons/pbar into the Tev
<bunch>         desired bunch number
options:        for PROTONS -> delay in secs; this is a modulo 40 count
                for P-BARS  -> A:R264F1 setting
                for P-BARS  -> event protection allows not triggering event $9A
```

FIGURE 2 On Line Help for Inject Command

The user can build a command by using the edit mode without memorizing the syntax of every command. The sequencer uses a fairly structured grammar. The main part of a command is the verb. There are currently 56 possible verbs that the sequencer recognizes. Using "INJECT" as an example command, because its on-line help is displayed in figure 2, one first selects the verb "inject". The object is chosen next, and in our example, either Protons or Antiprotons can be selected as the object of the inject command. Modifiers follow the object. In this inject example, a bunch number is the modifier. This modifier defines where in the Tevatron bunch train the beam will be injected. Some commands require more information, known as options, that are not displayed with the command on the screen. An example of a command with an option is a Plot command. The options contain information about which parameters are plotted, which graphics screen will be used and other details about how the plot will appear. The user building a command is stepped through its construction by menus, so the syntax will be proper even if the user has never before built a command.

The user can also build a type of complex command without the help of the programmer. There are sequencer files that the user can work from within the sequencer program. A file can be built to contain many individual tasks that are executed as a single file command.

MODES

There are 11 modes or instances of the sequencer. Each mode is actually the same program, but contains a unique set of commands. One mode of the sequencer is the Antiproton mode. This mode is used by the operator preparing the Antiproton source for a transfer (or preparing the source for Antiproton production). The Collider mode is used by the operator preparing the Tevatron for a transfer. Once all accelerators are ready for a transfer, the Collider sequencer becomes the master, orchestrating the rest of the fill process. Other accelerators have a unique mode of the sequencer for obvious functions. There are also a few other modes that have grown out of need, such as a studies mode which is used by physicists who want to prepare the execution of a study. There is also a backup mode which is used to periodically archive the state of the sequencer. The backup mode can then be used to view or operate the collider as it existed when the archive was made.

WEAKNESSES OF THE SEQUENCER

There are times when we want to store Protons in the Tevatron, without injecting any Antiprotons. This requires that the operators skip any step that was designed to transfer Antiprotons from the Antiproton source to the Tevatron. This goes against the single threaded design of the sequencer. The way this exception is handled is with a text message that appears at a certain point in the sequence. The message instructs the operator how to break the sequence safely if proceeding without Antiprotons. This technique is slightly awkward, but is a usable approach. This weakness in the sequencer was actually designed in to keep the operation procedure more structured and simple.

FLEXIBILITY AND TRIUMPHS

During a recent studies period devoted to upgrading the collider for operating at a higher luminosity, the sequencer was an invaluable tool. The upgrade hinged on injecting more bunches of Protons and Antiprotons into the collider. Many parameters are different for the multibunch mode of operation. There are several extra beam manipulations required once the beam is in the Tevatron. The extra complexity of the operation was handled remarkably easily by the sequencer with minimal changes by the programmer.

An archive was made of the existing collider sequencer, and then the backup mode was edited for multibunch operation. Almost all of the changes required to the actual sequences could be made by the physicist in charge of the study by using the on-line editor. The only change the programmer had to make was to allow a new range of modifiers in the inject commands. Using the two modes of the sequencer, it was trivial to change from the existing way of operating the collider to the new commissioning state. The operators could switch to the other mode of operation simply by switching to the appropriate mode of the sequencer.

Other studies have also benefited greatly from the flexibility of the sequencer. As an example, during a past Fixed Target run, a physicist wanted to perform a machine study with beam stored at the injection energy. Putting the Tevatron into a storage mode during Fixed Target is very unusual, and management worried that returning to the physics program would be difficult and that a major disruption would result. Going into and out of the storage study was built into a sequence in the studies mode. Less than one minute before the study was to end, the complicated sequence to return to HEP was initiated with one interrupt. Beam was being extracted to the experiments one supercycle later. Since this change back to fixed target took 57 seconds and was done in a completely repeatable way, it was easy to get this study approved a second time.

The sequencer has also allowed the collider to be more efficiently utilized. A study was planned to measure the emittance growth of Protons at the injection energy. This measurement was needed, but there was no plan to take time away from the HEP program to collect the data. A studies sequence was built to orchestrate the measurement. The sequence remained unused for over a month until late one Saturday evening there was some time available when Antiproton transfers could not be done for a few hours. The crew chief consulted with the Tevatron coordinator about how to utilize the Tevatron during this available time and was asked to start this studies sequence. The first command was a written set of instructions about the necessary state the Tevatron. The state was described by what collider sequence should have been last executed. The rest of the commands then set up the measurement instruments, and the programs that would collect the data. The data collection then began and the study was completed before the studier arrived at the Lab.

CONCLUSION

The sequencer is one of the most heavily used and well respected application programs at Fermilab. This program is the main software tool used to operate the collider. There is a tendency to actually overuse it as it is easy to cover-up weaknesses with many systems with some clever sequence editing. The sequencer has become a part of the language used in communicating about the collider. Most discussions of events include what part of the sequence was executing. Most anomalous occurrences during collider operation, if not hardware failures, are a result of a user choosing to do things outside the sequencer. This is confirmation that the philosophy used in developing the sequencer was a sound principle.

REFERENCES

[1] R. P. Johnson, Personal communication, 26 October, 1995.
[2] R. W. Goodwin and R. P. Johnson, "Coordinated Control of the Energy and Time Dependent Parameters of the Tevatron," Proceedings Of The 12th International Conference On High-Energy Accelerators, pp. 594-597, August 11-16, 1983.