

AUTOMATING ELETTRA OPERATION WITH "ONE BUTTON MACHINE"

D. Bulfone, F. Potepan*, C. Scafuri

Sincrotrone Trieste, SS 14 - Km 164.5, 34012 Basovizza, Trieste, Italy

Abstract

ELETTRA has reached good operational stability. Repetitive tasks, like the injection procedure, can be automated in order to maximise the beam time delivered to users. One button machine is a task spawner that executes different tasks concurrently, following a pre-defined control flow logic. It uses conventional UNIX routines for process execution and communication, and a Motif interface. A description of its various features is presented, covering operator intervention during flow execution, error recovery and step-by-step help when manual operation is selected. Operational results are given.

1 INTRODUCTION

ELETTRA has provided synchrotron radiation to the users since the commissioning year 1994. The set of control room tasks that drive the machine to a well-known state suitable for user operations is fully defined and tested. The One Button Machine (1bm) Project has been specifically developed in order to automate such procedures, simplify the operator's job and increase the beam time delivered to the users.

Each procedure is decomposed into a series of "actions" dealing with a specific machine subsystem operation. Actions are implemented by dedicated software "modules". A manager, called 1bm, regulates the execution of the various modules and interacts with the operator through a graphical user interface.

2 THE 1BM PROGRAM

1bm [1] [2] is a programmable UNIX task manager that spawns, communicates and controls the logical flow of a predefined set of processes. Its user interface is based on the standard Motif look and feel, which is already adopted for the ELETTRA control room programs.

2.1 The Programmable Module Manager

The term "module" will be used throughout this article synonymous with process, task or application program, referring to any executable spawned by 1bm. The execution of a module is usually allowed after the successful termination of a specific set of modules. The logical constraints are expressed by means of simple boolean expressions based on the module termination result. By using the negated values in the expressions specific modules can be triggered in case of

failure conditions. This can be exploited by error recovery schemes.

An important feature of the manager is its capability to spawn many modules concurrently in order to reduce the overall time needed by a procedure.

The manager spawns the modules according to the rules written in a configuration file. This file, which implements a given procedure, is modified by means of a normal text editor but only the operations officer has the privileges necessary to modify it.

1bm acts as an independent "finite state machine" for each spawned module. Fig. 1 is a representation of its state transition diagram. Shaded arrows denote manual operator intervention. The following states are defined:

- INITIAL: it is the initial state of every module, i.e. a module that was never executed before.
- FROZEN: module execution has been formerly blocked by the operator.
- RUNNING: the module is being executed for the first time.
- ERROR: the module execution failed and terminated with an unrecoverable fatal error.
- WARNING: the module execution failed and terminated with a recoverable error.
- DONE: the module was successfully executed.
- RECOVER: the module is being re-executed after a recoverable error.

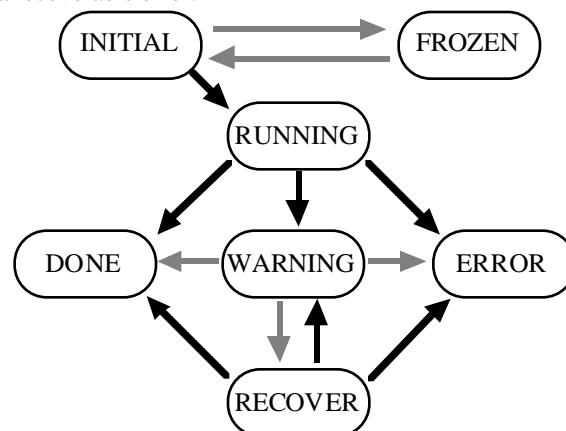


Figure 1: 1bm module state transition diagram.

2.2 The User Interface

The user interface, which is based on the standard Motif libraries, is well decoupled from the modules and not influenced by their execution or failure. Only data necessary to perform routine operations are shown by default, while details can be requested if necessary. The list of modules foreseen for a given procedure is initially displayed (fig. 2). Each module has a colour coded label that shows its current status, during

* now at Integrated Computer Solutions Incorporated, 201 Broadway, Cambridge, MA 02139.

execution. The panel also contains the main controls of 1bm: two buttons to formerly "freeze/thaw" the execution of selected modules, a button to start the whole procedure and a panic button to force the immediate termination of a module or all of them.

The following additional information panels are available for each module in the list:

- a window with a brief description of the parameters passed for the execution of the module and their actual values.
- two scrolled message windows showing, respectively, the redirected standard output stream from the module and a special subset of error messages. These are used to closely follow the execution of the module.
- a help window containing the information needed when manual operator intervention is eventually required. Such information is defined in the configuration file of the given procedure. There are also three buttons on the lower part of the window that force the manual transitions foreseen in case a module is in the warning state.

3 MANUAL OPERATOR INTERVENTION

Manual intervention of the operator is requested when module execution fails, but no fatal error has been detected. In this case the module switches to the warning status. The operator detects this condition and, with the help of the corresponding information window, attempts to manually terminate the module's action by means of the standard programs available on the control room consoles or by direct intervention on the equipment. On the base of the achieved result the operator eventually forces a module state transition to the correct current status (DONE or ERROR) or re-execute the module. The 1bm programmable manager then automatically continues or terminates the on

going procedure depending on the logic specified in the configuration file.

4 1BM MODULES

Any UNIX executable, binary object or shell script, can be a 1bm module so that already developed and well tested control room programs are easily integrated and re-used. Only four simple rules are adopted:

- modules must be based on standard UNIX conventions, libraries and system resources;
- module's execution parameters are taken from the command line;
- any information or error message must be printed on the standard output;
- modules must terminate clearly, returning to the calling environment a defined exit code indicating the success or failure of their execution.

4.1 Special Strings

1bm recognizes some special strings sent by a module to the standard output and displays them on a dedicated section of the associated message window. This feature is useful in case of problems to quickly identify what is going wrong. A set of definition files and a module skeleton making use of such special strings is available for writing new modules. The use of special strings is however absolutely not mandatory for a module to be integrated in 1bm.

5 INJECTION PROCEDURE

The injection procedure is performed once a day during "user shift" periods. The operator has first to dump the beam, open the insertion devices (ID) gaps, close the front-end shutters and valves, set the injection elements, cycle the magnets and load the machine files.

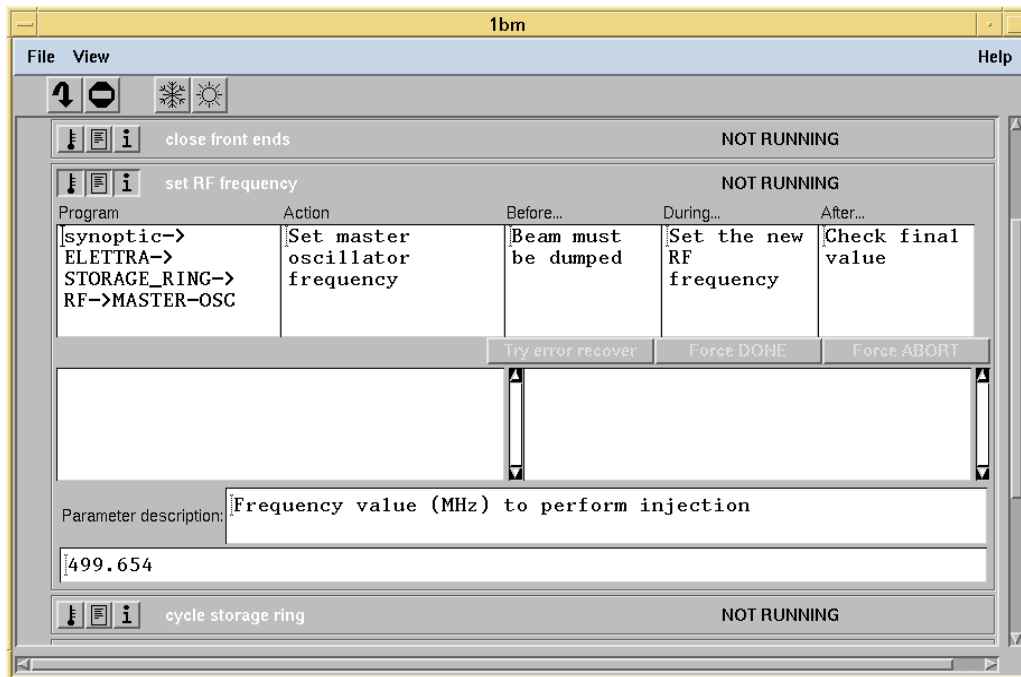


Figure 2: 1bm Motif user interface.

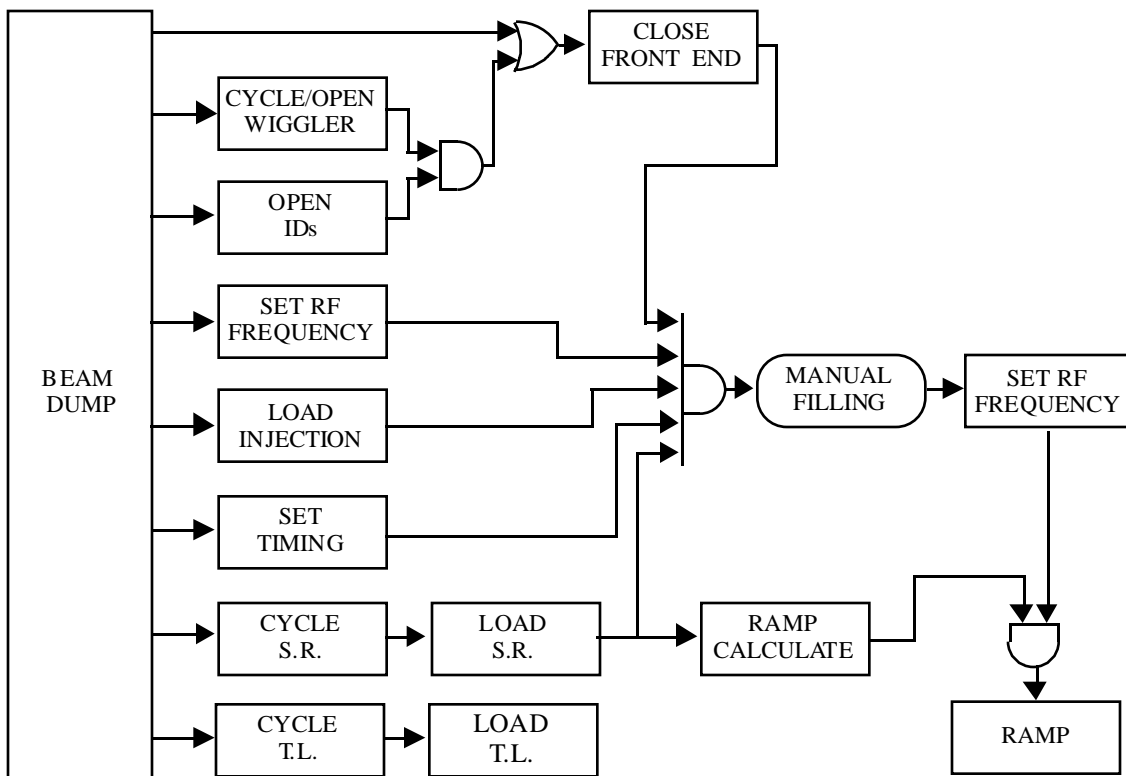


Figure 3: ELETTRA injection procedure logic flow diagram.

After filling the ring with the requested current value, he has to execute a ramp to the final energy, close the ID gaps, correct the orbit to zero position and slope at the centre of the IDs and open the front-end shutters and valves to provide light to the users.

The injection procedure has been almost completely automatized by 1bm [3]. The implemented logical flow involving many concurrent tasks is shown in fig. 3.

The average time spent from "beam dump" to "manual filling" has been reduced from about 40 minutes to less than 20 minutes. This time cannot be reduced further since it is fixed by equipment hardware constraints. The main duty of the operator is to fill the ring with the requested current value.

Some newly developed 1bm modules are now under commissioning in order to automatize also the energy ramping of the ring after the manual filling. First results are quite good as the total injection procedure time, including manual filling, has been cut down to 26 minutes.

6 CONCLUSIONS

1bm has been preliminarily installed in the ELETTRA control room since mid 1996 and has been routinely used to perform the injection procedure since the beginning of 1997. More and more modules are being integrated and the preparation of the machine for user operations has been considerably speeded up. Injection is carried out by a single operator and the rate of human errors has been minimized. The adoption of a well defined configuration file guarantees that the

machine is always operated with the desired parameters and increases reproducibility.

The spectrum of applications for 1bm is not restricted to machine operation. It can be successfully used every time a logically ordered execution of related modules is required, just specifying a different configuration file.

Future extensions deal mostly with the automatic error recovery: a simple convention should be defined when error messages are sent from a module to 1bm and a smart error recovery performed by the Programmable Module Manager. Only unsuccessful commands will then be performed in order to save the time needed for a complete re-execution of the module.

7 ACKNOWLEDGEMENTS

We would like to thank the ELETTRA Operators and the Operations Group for their useful comments and suggestions.

REFERENCES

- [1] F. Potepan: 'Automize Machine Operation at Control Room Level: One Button Machine Program', ST/M-TN-96/3, Sincrotrone Trieste, 1996.
- [2] D. Bulfone et al.: 'Controls in the Past Year of ELETTRA Operation', Proc. Int. Conf. on Accelerator and Large Experimental Physics Control Systems, Chicago, Illinois, 1995.
- [3] C. Scafuri: 'Automatic Machine Operations', Proc. 4th Annual Workshop on European Synchrotron Light Sources, Trieste, 1996.