# TROUBLE: A TROUBLE SHOOTING PROGRAM FOR ELETTRA

E. Karantzoulis, C. J. Bocchetta, M. Plesko+ and L. Tosi, Sincrotrone Trieste, Italy

## ABSTRACT

Trouble is a diagnostic system entirely developed at ELETTRA that guides the operator to diagnose and repair in a fast and efficient way a machine fault. The program is written in C and it is installed on the ELETTRA control work station. The method used and its performance will be presented and discussed.

## 1 INTRODUCTION

ELETTRA, the 2 GeV third generation light source in Trieste -Italy, has entered its third calendar year of operations and is routinely delivering photons of very high brilliance in the VUV to soft X-Ray region. It consists of a linear accelerator operating at 1 GeV, a transfer line and a storage ring [1] that also performs energy ramping to 2 GeV.

Once the initial part of the commissioning i.e. from October 1993 to June 1994 was concluded it was decided to record in a written form the accumulated knowledge on problems concerning the machine, its behaviour under various circumstances and the recommended remedies.

The first form of that "recording" was a simple list of written instructions repeatedly controlled, modified and enlarged in order to speed up the required intervention of the experts in case problems arose.

It was soon realised that such a list incorporated in the main control system could in parallel be used as a very efficient trouble-shooting tool with many advantages during machine operation for the following reasons:

- during user runs, operators and non experts operate the machine.
- operators will learn from the accumulated knowledge.
- the intervention time to trouble-shooting is reduced since the operator, by correctly recognising the problem, acts in the most efficient way.
- new operators know little of the machine equipment and therefore need help.
- all operations in ELETTRA are made from the workstation terminals, therefore operators find an on-line help and trouble-shooting system very convenient.
- other useful information relative to security, operative administration etc. may also be incorporated.

Thus ELETTRA's trouble-shooting system can be seen as a diagnostic system. Expert knowledge is encapsulated in a series of if-then clauses. The user is presented with a question and several possible answers. The choice of an answer leads to a new question and so forth, until the final solution is reached.

The software with the trouble-shooting knowledge is divided into the knowledge database which contains all the knowledge of the experts and the program which leads the user through this information in search of the answers.

## 2 THE PROGRAM

The program consists of a parser, a run-time data structure and a simple user interface. At start-up, the parser interprets the knowledge database and builds up the run-time data structure which contains the questions, answers and links between them. The user interface guides the user through this structure.

### *2.1 The Parser*

In order to facilitate the entry of expert knowledge, a simple description language has been defined. Each question in the knowledge database has to be defined in the following structure:

```
Question_name
 {
 text of the question
 can span several lines
 ...
 }
 Answer_1: text of answer 1
 Answer_2: text of answer 2
 ...
 Answer_n: text of answer n
 ...
```

The identifier `Answer_n` is the name of another question, which will be asked after this answer has been selected by the user. Each question can have any number of answers. If no answer is listed, the program assumes that the text of the question `Question_name` contains actually the final explanation which leads to no further question.

The somewhat redundant formalism with brackets, semicolons and full stops is necessary as it helps the parser which reads the database line by line to spot syntactically wrong constructions and inconsistent relations between questions and answers. The parser checks for duplicate question names, question names which appear as answers but are never defined and questions that are defined but never appear as possible answers.

### *2.2 The Run-time Data Structure*

The parsers allocates each question encountered to the run-time data structure which is built up with the same tools as our other high level software [2]. This structure represents a directional graph in the mathematical sense: Each question is a node in the graph with some paths leading to it and other paths leading from it.

The only reason for keeping the data in the run-time structure instead of letting it remain in the file is to speed the operation. Adversely an indexed database file is not necessary, as the amount of data is sufficiently small and can be easily kept in the memory.

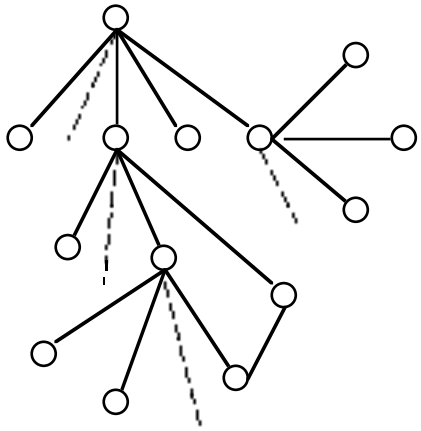
### *2.3 The User Interface*

Once the knowledge database has been parsed, the user is presented with the first question and several possible answers. The choice of an answer leads to a new question and so forth, until a final solution is presented. In case the user realises that the new question does not lead in the right direction, (s)he can undo the last answer. An unlimited number of 'undo' operations is supported.

Currently the user interface is limited to a text-only version. However, it already contains all necessary features for a simple and user-friendly operation.

## 3 THE KNOWLEDGEABLE DATA BASE

The database structure is a multiple tree structure allowing also for inter-branch connections.



Experts were asked to provide in an "if-then" form what they knew about problems concerning the machine i.e. hardware, behaviour etc. and the needed actions for repairing. The writing in was not an easy task however becomes easier when one gets acquainted with its special structure. The parser checking for duplicate names, question names that are not defined etc. provides an additional help to the correct and fast writing.

Other information except trouble shooting is also stored like the injection procedure, the start up procedure and general information about operations.

## 4 AN EXAMPLE SESSION

In a real case, after injecting 250 mA the lifetime was very low. The operator found the solution to this problem with Trouble through the following session:

**WELCOME TO ELETTRA EXPERT SYSTEM**
* MENU *
************************************************

1: shut down inf.
2: Procedure for a good start-up
3: The current injection procedure
4: **the actual trouble shooting part**
5: general information about the machine operation
Choose from 1 to 5, ctrl C to quit,'-' to go back, then press RETURN
Your answer was: the actual trouble shooting part

(here the operator selects the trouble shooting part.)

---

Is there a stored beam?
1: no
2: **yes**
3: maybe
4: got problems with the Experiments
5: got problems with Linac
Choose from 1 to 5, ctrl C to quit,'-' to go back, then press RETURN
Your answer was: yes

(other possibilities exist too to deal with other problems, the choice obviously here was that there was already stored beam.)

---

Why you call for help if there is beam?
1: in fact, there is no beam and you miss typed the answer
2: OK, there is beam and I want to quit TROUBLE
3: the beam quality is bad!
4: the beam orbit is abnormal
5: **the lifetime is low**
6: injecting current saturates
7: slow injection rate
8: not inject !
9: can not ramp
10: experiments complains of oscillations/spikes no visibly seen on synchrotron light profile monitor
Choose from 1 to 10, ctrl C to quit,'-' to go back, then press RETURN:
Your answer was: the lifetime is low

(here the operator selected the low lifetime point.)

---

Probable causes- control in the written order:

1) **verify ring filling**-> look at the oscilloscope and control fillpat
2) verify that no scraper blades are in !!!
3) verify orbit
4) verify vacuum
5) verify tunes
6) verify harmonic sextupole
7) verify sextupole values/ chromaticity
8) control the MBIs- call for real experts

1: exit

Choose 1 to continue, ctrl C to quit,'-' to go back,
then press RETURN: 1
Your answer was: exit

(Actually the filling of the ring was mistakenly set to 10% i.e. 250 mA in 43 bunches and consequently the lifetime was low.)

## 5 CONCLUSIONS

The system works well, is reliable and is very useful for less experienced operators. More expertise is added into the database when something new emerges. However a graphical interface is missing. To simplify our work, we plan to use the hyper textual features of WWW-browsers which have a good graphical interface with which most of our staff has already had experience. The knowledge database will be composed of several hypertext mark-up language (html) files, one for each domain. The new structure will provide on-line images, drawings and documentation in Postscript format, as all of this is already supported by WWW-browsers.

## REFERENCES


[1] ELETTRA - *Conceptual Design Report*, Sincrotrone Trieste, April 1989.

'*ELETTRA Commissioning and Operation*', by C. J. Bocchetta et al., Proc. EPAC 94, London 1994, (1995), 579-581.

[2] '*A Complete Data Structure for the High Level Software of ELETTRA*',by M.Plesko, Proc. EPAC 94, London 1994 (1995), 1773-1775.