The High Level Software of ELETTRA

M. Plesko, C.J. Bocchetta, F. Iazzourene, E. Karantzoulis, R. Nagaoka, L. Tosi
Sincrotrone Trieste
Padriciano 99, I-34012 TRIESTE
e-mail: plesko@elettra.trieste.it

Abstract:

ELETTRA is a third generation light source at an electron beam energy of 1.5 to 2 GeV. For commissioning and operation a multitude of high level applications have been prepared and successfully used which are presented in this paper together with a simulation program which can be linked to the applications instead of the control system in order to thoroughly test and debug the applications before real operation. The applications use a unique data structure and a set of routines which transparently access the control system. All these approaches significantly reduced development time and ease future maintenance.

I. INTRODUCTION

The third generation light source ELETTRA [1] consists of a linear accelerator, a storage ring and a 100m long transfer line between them. Both the transfer line and the storage ring are attached to a control system, which has been developed in-house[2].

Modern accelerator control systems like the one of ELETTRA allow digital access to practically all points of the accelerator. Most of the machine physics related tasks that were previously done manually and analysed off-line can now be automated via software which has direct access to the controlled points through the control system. The software which performs these tasks is generally called high level software (HLS).

Another definition of the high level software is that it is software which does more than just simple settings or readings of one controlled point. It either manipulates the data it gets or acts on several different elements and is thus not solely a control system action. The topics of typical HLS applications include measurements and analysis of measurements, orbit correction, modelling and machine monitoring.

The HLS which was used during the commissioning and later for the operation of ELETTRA had been defined well in advance [3]. It was ready and tested on "day one". This had been achieved on the ground of a common data structure which was used by all programs and a simulation program with which the applications were tested and debugged before they were used on the real accelerator.

The unique data structure which provides all relevant machine physics parameters readily in the data structure or on a function call was used in order to have a standardised approach thus simplifying maintenance of the HLS.

The simulation program intercepts calls to the control system and executes a model of the machine instead. All misuse of the control system is thus spotted immediately and can easily be corrected.

II. THE HLS DATA STRUCTURE

The data structure contains all data which describe the transfer line and storage ring from the machine physics point of view and in addition contains all necessary references to the control system in order to access single and multiple controlled points. In this way it greatly simplifies the development of applications. The content of the structure is not specific to ELETTRA, since it is generated dynamically at run-time from several text-only editable data files. Due to this generality, it can be easily ported to any accelerator complex.

The data structure consists of three components: **the fixed data files**, the **run-time data structure** and the **utility routines**. The data files contain the calibration data and other fixed parameters of the accelerator elements. The run-time data structure keeps these fixed data and other data that is read in from the control system and converted to machine physics parameters, such that it is easily accessed by the HLS program. The utility routines allow transparent access to single components or groups of them, store obtained data in the run-time data structure and provide modelling tools. The linear model of the machine taking into account the actual magnet currents is generated by a call to a single routine. The HLS data structure is described in detail in a separate contribution [4].

III. THE SIMULATION PROGRAM

GAP (Ghost Accelerator Program) [5] is a code which simulates an operating accelerator. The program performs mainly two tasks : *modelling* of a real accelerator lattice and *simulation* of the ELETTRA control system response to application programs. The code is made so that the application programs can run either with the real control system machine or GAP without any change.

The model resembles as closely as possible the machine ELETTRA. The code models both transfer line and storage ring. The model contains various elements, including the power supply families of the magnets and diagnostic devices, like the beam position monitors (BPM), fluorescent screens, scrapers, etc. Each magnet has its calibration coefficients which map the strength to current and current to strength [6], like in the application programs. The tracking of the particles is performed for both first turn and stored beam mode. To produce realistic diagnostic information, the vacuum aperture chamber of each element of the accelerator is included in order to detect the particle loss and where it occurs. Errors can be assigned to the elements of the lattice, like misalignments and magnetic field errors. The multipole components can be from dipole to 20 poles, normal and skew. The code also simulates faulty elements, e.g., magnets wrongly connected to their power supply, or magnets not connected to the right power supply, wrong calibration coefficients, etc.

The formalism treats the septum, bending and quadrupole magnets using linear transformation matrices. While the sextupole, corrector and kicker magnets and the misalignments and magnetic field errors are instead treated using the thin lens approximation based on a Taylor's magnetic field expansion. The code is written in FORTRAN. Error messages from the control system are also simulated. For example, exceeded minimum or maximum current for setting or ramping magnets, no beam at a diagnostic device, etc.

Beside these features, GAP also prints out data and messages which significantly helped in debugging and testing of the application programs. Furthermore, GAP has been used during commissioning as a simulator to help understand the behaviour of the machine.

IV. THE MACHINE FILE

The machine file allows to uniquely save and restore the status of the machine - the transfer line and the storage ring in the case of ELETTRA. The machine file is used to restore the equipment which is relevant for machine physics and machine operation. It therefore contains the currents of all magnets (bending, quadrupole, sextupole and correctors), the insertion device gaps and the injection element settings. Both the set values and the actual read values are recorded.

Thus the machine file can be considered to be a snapshot of the machine. All HLS applications can access either the actual machine directly or read in the machine file which represents the machine at another instance. Conversion routines convert the currents from the machine file into machine physics variables and vice versa.

V. THE APPLICATION PROGRAMS

All application programs are written in the computer language C. The decisive reason was that the graphical interface used by the programs is based on the X11/Motif tool kit, which is supported in C only. The programs run in a UNIX-like environment on Hewlett-Packard 9000/700 series work stations.

Great emphasis has been placed on the graphical user interface of each program. The interaction with the user is completely event driven. The power of X-Windows and the Motif window manager is exploited by utilising buttons, pull down menus, sub windows, etc. Graphical output is presented in a special plot-widget [7], which has been developed in-house.

The applications are running on the normal control system workstations. However, due to their distinct functionality, they are invoked and run independently of the control system's man-machine-interface.

The following subsections describe the most important programs, based on their purposes.

A. Measurements, Analysis and Modelling

The emittance of the linac is measured in the transfer line with the program **Emittance**. It varies one or several quadrupoles and determines the emittance and Twiss functions at the linac exit from the change in the beam profile. The beam size variation with quadruple current and the resulting phase space ellipses are displayed graphically.

The program **Tune** [8] has been written mainly for applying desired tune values to the storage ring. During the setting of the tunes, the user may also record the associated lifetime and beam current and perform a correlation between these parameters and the tunes currently being set by the program, giving thus a measurement of the stop bands of any encountered resonances.

Optiks [9] is a program that measures and corrects the optical functions (i.e. beta functions, phase advances, tunes, dispersion and chromaticity) in a circular accelerator. The program offers in a user friendly and compact manner almost all the needed information that the operator wishes to know about the machine optics as well as the means for correcting. It makes a wide use of the high level software data structure including its build-in Twiss functions calculation that permits to calculate in a fast and easy way the nominal machine optics. The measured machine optics functions are obtained by analysing the closed orbit data. Alternatively the quadrupole strength shifting technique is also employed to measure the beta function at the quadrupole locations. Independently the user may also measure and correct the dispersion and chromaticity.

The integer part of the tunes is determined by a Fourier transform of the orbit. One can also get the Fourier transform of the corrector strengths. Since the routine does not need a closed orbit, it may be used for injection studies and it was particularly useful at the initial stages of the commissioning. This routine under the name **Harma** is also a separate program.

B. Orbit Correction, Optics Correction

There are two HLS programs developed for the correction of orbit in ELETTRA. One is **Orbit** and the other is **Bump** [10]. The program **Orbit** integrates all necessary actions on the orbit of both the storage ring and the transfer line together with many options. Its core is the orbit correction software package *COCU* developed at CERN [11]. *COCU* is equipped with well known correction schemes such as MICADO, which utilises an unique data structure management system *MOPS*. Apart from some extensions made on *COCU*, a major effort was therefore made in the construction of the entire structure which embeds *COCU* and *MOPS* in a way optimised for the machine operation. The graphical user interface plays the central role binding different parts including the run-time data structure that interfaces the machine. All operations including those required for *COCU* are handled by simple manipulations of graphical objects (called widgets) on the control panel.

The program **Bump** on the other hand is a dedicated program, developed separately to create local bumps in an interactive manner. The main characteristics of the program are:

- the continuous display of the orbit interpolating the consecutive BPM readings taking into account also the corrector kicks by numerically fitting the trajectory;
- the graphical scroll bars with which the user can arbitrary select the bump location and at which the amplitude and the slope of the current orbit can be set;
- the representation of magnetic elements in terms of graphic push buttons with which the desired correctors can be selected.

All operations can be entered in parallel for the two transverse planes. The program is particularly optimised for local corrections of orbit in the insertion devices.

The program **IDcomp** [8] has been developed in order to compensate the linear distortions due to insertion devices. The software presents several compensation techniques, which go from local compensations, where the distortions are forced to be localised, to global compensations, where the beta beat around the ring is minimised. Tune re-adjustments are also possible. An important feature of the software is the possibility for the user to visualise the resulting optics, before any action is performed on the machine.

C. Machine Monitoring

The program **MonitorBPM** displays the readings of all 96 BPMs in the storage ring as a function of time, thus monitoring the drifts in the orbit. In four different display windows, the average and the r.m.s. values of each BPM are plotted for the respective planes. The programs saves the entire data collected into a file for post processing.

The program **Mramp** displays the beam energy which is defined by the dipole current, the beam current, the betatron tunes and the average and r.m.s. values of the orbit as a function of time so that the operator can monitor the beam behaviour during the course of the energy ramping. As in the previous program, the collected data are saved into a file when exiting from the program.

The programs **MonitorPS** and **Hist** track the behaviour of the power supplies. Both the set current from the control system and the actual current read from a current transformer are recorded for each power supply. The difference between the set and the reading is compared and displayed graphically with **MonitorPS**. **Hist** histograms the difference between the reading at start-up with the actual reading over longer periods and issues alarms if the difference exceeds the design values.

D. Miscellaneous Tools

Display is an interactive optics design tool. The user may visualise through graphs and numbers the optics associated with the actual power supply settings. By zooming into one section, the user sees a layout of the elements and may change interactively in the model the quadrupole settings and the insertion device gaps and visualise the resulting optics.

The programs **EnergyTL** and **EnergySR** scale the optics of the transfer line and storage ring, respectively, to another energy set by the user. Those programs were extensively used during linac and transfer line commissioning and storage ring first turn steering and initial accumulation.

The **PScyc** and **PSon** programs are low level applications, which cycle all or a set of magnets and switch them on or off according to a requested procedure. Eventually, these routines will evolve into an automatic start-up program.

VI. CONCLUSIONS

The high level software was used very successfully during the commissioning and first operation of ELETTRA. Most of the results of ELETTRA presented at this conference were obtained with the help of the HLS (see [12] and references therein).

All major HLS applications exist already. New are being developed whenever the need for additional actions arise. Due to the well defined environment of the data structure and the simulation program, it is relatively easy to implement new programs in very short time. Most of the applications can be transferred to other accelerators as long as their control system uses UNIX workstations. The use of standards (C, X11/Motif) and transparency in the access to the actual control system implementation insure this.

VII. ACKNOWLEDGEMENTS

The authors thank the members of the ELETTRA control group for tremendous help on the X11/Motif tool kit and other computer related topics. Further thanks for fruitful discussions go to R.Richter and A.Wrulich.

VIII. REFERENCES

[1] ELETTRA - Conceptual Design Report, Sincrotrone Trieste, April 1989.
[2] D.Bulfone, *Status and Prospects of the ELETTRA Control System*, Proc. ICALEPS 93, Berlin 93.
[3] M.Plesko, *The High Level Software Routines*, Sincrotrone Trieste. ST/M-TN-91/11. 1991.
[4] M.Plesko, *A Complete Data Structure for the High Level Software of ELETTRA*, these Proceedings.
[5] F.Iazzourene, *GAP: Ghost Accelerator Program*, Sincrotrone Trieste. ST/M-94/.. 1994.
[6] F.Iazzourene, *Calibration Coefficients of ELETTRA Magnets*. Sincrotrone Trieste. ST/M-TN-93/11. 1994.
[7] F.Potepan, Plot Widget 1.6, Sincrotrone Trieste, 18.5.1994.
[8] L.Tosi, *The High Level Programs: Tune and ID-COMP*, these Proceedings.
[9] E.Karantzoulis, *Optiks: An Optics Measurement and Correction Program for ELETTRA*, these Proceedings.
[10] R.Nagaoka et al., *Orbit Correction in ELETTRA*, these Proceedings.
[11] B. Brandt et al., *COCU User Guide (Version 7.0)*, CERN-SL/AP/Note91-19, 1991.
[12] A.Wrulich, *Status of ELETTRA*, these Proceedings.