

ELETTRA NEW FULL ENERGY INJECTOR HIGH LEVEL SOFTWARE

C. Scafuri, F. Iazzourene, Sincrotrone Trieste S.C.p.A., Trieste, Italy

Abstract

The Control system for the new full energy injector [1] will be entirely based on Tango with an Object Oriented distributed architecture. The availability of the new and modern software platform led us to design a new High Level Software framework for model based accelerator control. The new design is fully Object Oriented and follows a modular approach. The main modules provide a set of different views or abstractions of the underlying accelerator: field module, machine description module, optics module. The field module handles all the access and communications with the actual devices of the accelerator, e.g. power supplies, instrumentation, etc. The field module also manages the conversion from actual values acquired from the field into values meaningful to electron beam dynamics. The machine description module handles the machine lattice and layout description. The optics module performs the calculations relevant to application programs like the Twiss functions. An important characteristic of the new library is the unified management of all the needed calibrations and configurations by means of a relational data base.

INTRODUCTION

Tango [2] is a distributed control system with a full object oriented architecture. The new High Level Framework (HLF) will provide robust tools for the development of application programs which will be needed for the commissioning of the new injector, for machine physics measurements and later for routine control room operations. A typical use of the HLF based program will be the manipulation of the tunes, chromaticities, orbit, dispersion, etc. In order to perform these tasks the HLF will contain a set of accelerator modeling classes. The new HLF is designed to be used also for the Elettra storage ring, so it will include models for the insertion devices and it will handle both circular machines and transfer lines. In future it will also be used for the operation of the transfer line and FEL undulators of the FERMI@Elettra [3] linac-based free electron laser.

MODULAR OBJECT ORIENTED DESIGN

The new HLF design is fully object oriented and follows a modular approach (see Fig. 1 in UML notation [4]). The analysis of model based accelerator controls led us to split the HLF functions in 3 distinct areas, which are mapped to modules: optics calculations, management of configurations and calibration, access to the actual settings of the machine and unit conversion. The different modules group together classes and functions performing strictly related

tasks. The various modules are weakly coupled: a module access the services provided by another module only through a set of well defined interfaces. The internal design and implementation of the module classes can be changed without affecting the other modules. The framework is written in C++ and exploits as much as possible the Standard Template Library and algorithms.

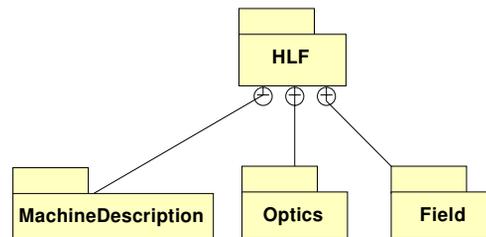


Figure 1: HLF modules.

OPTICS MODULE

The optics module, nicknamed Vicky, packages all the classes and procedures needed by application programs to analyze and manipulate the accelerator from a machine physics point of view. It makes use of the services provided by the other HLF modules.

Object Model

The object model closely follows the machine physics point of view of the accelerator. A set of classes, deriving from AcceleratorComponent, is used to model the characteristics of the accelerator building blocks (see Fig. 2). Another set of classes handles the actual assembly of AcceleratorComponents into an actual accelerator (see Fig. 3). The Section class is introduced to map the common concept of accelerator section or cell, rather useful for periodic machines like our storage ring. Similar object models are found also in other object oriented modeling packages [6], [7].

Machine Model Calculations

The mathematical model implemented by AcceleratorComponent classes is based on 6x6 transfer matrices, which are sufficient to describe the transverse coupled dynamics of a constant energy electron beam. In view of the utilization of the new HLF also for the Elettra storage ring, complete models for the insertion devices, including the exotic ones like Apple-II and figure-8 type, is foreseen. While the existing Elettra High Level Software library [8]

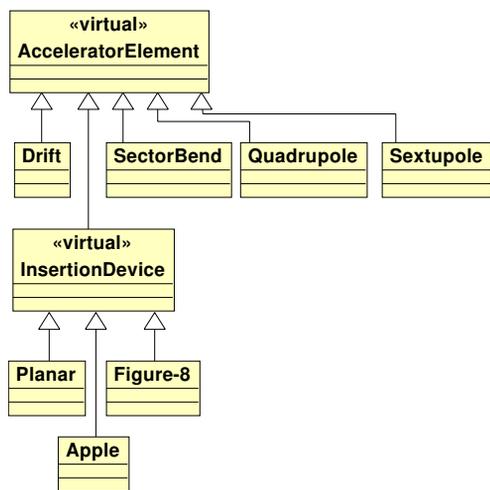


Figure 2: HLF Optics classes.

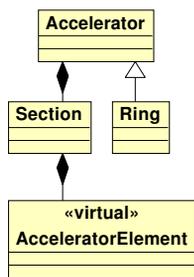


Figure 3: HLF Accelerator classes.

calculates only the optics of the ideal lattice, the AcceleratorComponent mathematical model supports also alignment and multipolar field errors. This will permit a better approximation of the real accelerator lattice and therefore provide more accurate measurements and corrections.

MACHINE DESCRIPTION MODULE

The machine description module gives access to the static data needed to build and initialize an accelerator model. This data can be divided in 3 different areas: lattice description, connection with control system devices and calibration of electro-magnetic devices

Lattice Description

The lattice description produces the list of the elements together with the data needed for its complete definition. The data is presented to the user in a standardized format, which can be easily used by the optics module classes. The standard method to access the lattice description will be by means of classes designed following the factory and builder pattern [5] which automatically instantiate the whole lattice structure. The initial lattice description will be stored either in a free format text file or in a relational data-base. The file based description is easier to edit and modify. The data-base description, if correctly administered, provides

better data protection. Moreover, due to the single centralized data source, guarantees the unambiguous definition of a given lattice. These characteristics are very important for control room operations.

Connection Description

The connection to control system devices is defined by associating the name of an AcceleratorComponent to the name of the relevant Tango Device Server. The Tango control system will automatically and transparently translate the Device Server name into the appropriate network address. The number of different Tango Device Servers found in the connection description gives us the number of effective degrees of freedom of the accelerator model. In our case this number is reduced by the fact that we have several magnets connected to a single power supply. This important structural information will be available from the connection description: given an AcceleratorComponent name we can ask for the list of all the AcceleratorComponents connected to the same Device Server. As for the lattice description, the connection description will be stored either in text files or in a relational data-base.

Calibration Tables

Calibration tables are mostly needed by magnetic elements. They contain all the available measured data for each element. The format of the data depends on the type or class of the element. For example the calibration table of a SectorBend will list a number of excitation currents and the corresponding nominal energy, field index and magnetic length. Due to the large amount of data the calibration tables will be stored exclusively in a relational data-base.

Interfaces

The Machine Description module presents to the user a set of interfaces. The purpose of the interfaces is to provide a uniform way to retrieve the wanted data, whether it is stored in text file or in a relational data-base. The user will only need to know the name of the source of data, all the low level text parsing or table queries are then automatically selected and performed by the relevant interface.

FIELD MODULE

The field module implements all the methods needed to get the status, settings and readings of any AcceleratorComponent object. In the case of the magnets, for example, it provides a direct translation of a current to a strength, and vice versa.

Access to Field

In order to perform these operations the Field module classes will interact with the actual hardware device by means of the control system. We will hide to the user all the non relevant details and make visible only what is

needed to actually perform machine physics operations. In the quadrupole magnet example, the HLF does not need to have a full knowledge of the relative power supply status, but only whether it is working or not. Configuration data of any element, like minimum or maximum of a power supply current, is directly available on line exploiting Tango facilities.

Mapping with Calibration Tables

The field module converts machine physics quantities, for example a quadrupole strength, to field quantities, current in the example. This is done with appropriate interpolation functions on the above mentioned calibration tables. Each class of magnet, including insertion devices, will use an appropriate interpolation algorithm.

On-Line Field Model as Tango Device Server

As already done at the SLS[9], we will implement an on-line field model for our machines. It will have the form of a standard Tango Device Server. This will allow us to use part of the HLF modules also with other simulation packages, such as AT[10], and to exploit advanced Tango features, such as event notifications, in HLF based machine physics programs.

REFERENCES

- [1] M. Svandlirk et. al., "Elettra New Full Energy Injector Status Report", this proceedings.
- [2] A. Götz et al., "TANGO: a CORBA-based Control System", ICALEPCS 2003, Gyeongju, Korea.
- [3] FERMI@Elettra, this conference
- [4] G. Booch, J. Rumbaugh, I. Jacobson "The Unified Modeling language User guide", Addison-Wesley, 1999, ISBN 0-201-57168-4
- [5] E. Gamma, R. Helm, R. Johnson, J. Vlissides, "Design Patterns", Addison-Wesley, 1998, ISBN 0-201-63361-2
- [6] H. Nishimura, "Goemon, a C++ Library for Accelerator Modeling and Analysis", PAC 2001, Chicago, U.S.A.
- [7] J. Galambos et al., "SNS Application Programming Environment", EPAC 2002, Paris, France.
- [8] M. Plesko et al., "The High Level Software of ELETTRA", EPAC 1994, London, England.
- [9] M. Böge, J. Chrin et al., "Development of Beam Dynamics Applications within a CORBA Framework at the SLS", EPAC 2000, Vienna, Austria
- [10] A. Terebilo, "Accelerator toolbox for Matlab", SLAC-PUB-8732, May 2001