COMMISSIONING OF THE SLS USING CORBA BASED BEAM DYNAMICS APPLICATIONS

M. Böge, J. Chrin, M. Muñoz, A. Streun, PSI, Villigen, Switzerland

Abstract

A distributed client-server model, based on the Common Object Request Broker Architecture (CORBA), has been established to interface beam dynamics applications at the Swiss Light Source (SLS) to essential software packages such as the accelerator physics package TRACY and the Common DEVice (CDEV) control library. Within this model remote clients can invoke computer intensive methods, such as beam orbit correction procedures, on a dedicated server. Access to the SLS accelerator devices is achieved through a generic C++ CDEV server.

CORBA based applications have been extensively used during the commissioning of the SLS booster and storage ring. It has been demonstrated that the complex client-server environment is manageable and reliable. The inherent flexibility and the relative "ease of use" of the established framework have justified the initial investment in its design and implementation.

Beam dynamics related commissioning results are presented together with the corresponding applications.

1 INTRODUCTION

The SLS [1] is a 2.4 GeV electron storage ring currently under commissioning [2] at the Paul Scherrer Institute (PSI), Switzerland. Electrons from an injector booster synchrotron, fed by a 100 MeV linac, are transferred to the storage ring at full operating energy. Scheduled for operation in August 2001, the SLS will provide synchrotron radiation of high brilliance to experimenters from a variety of disciplines. A considerable number of high-level beam dynamics Application Program Interfaces (APIs) are required for the commissioning and operation of the SLS accelerator complex and for machine physics studies. These APIs typically share a number of generic tasks including:

- access to an accelerator physics package,
- accelerator device control,
- database access and management, and
- logging of messages and alarms.

With the aid of object-oriented methodology, common functions can be identified and developed as reusable components. Furthermore, a distributed system allows optimal use of available resources, an important consideration given the CPU intensive physics algorithms employed by the accelerator modelling procedures. To this end, a distributed client-server model, based on the Common Object Request Broker Architecture (CORBA) [3], has been proposed [4]; client programs readily access shared services, either locally or across the network, through CORBA objects.

2 THE CORBA FRAMEWORK

In the evolution of object-oriented distributed computing systems, CORBA is a recent standard that provides a mechanism for defining interfaces between distributed components. Its most distinguished assets are platform independence, in so far as the platform hosts a CORBA Object Request Broker (ORB) implementation, and language independence, as ensured through the use of the Interface Definition Language (IDL). The latter feature is of particular interest to SLS beam dynamics API developers as it provides for the option between high-level application languages.

2.1 Server Hardware and System Software Components

The chosen platform for server software components is a 2x600 MHz Pentium III system ("Model Server") running Linux. A mirror server is permanently available to provide redundancy in case of a failure of the "Model Server".

The ORB (Object Request Broker) employed is MICO [5] a fully compliant CORBA 2.3 implementation available under GNU public license terms. Use is made of the Implementation/Interface Repository facilities, the Naming Service and the Event Service of the ORB. MICO provides IDL to C++ mapping. The Tcl interface Combat [6] adds the Tcl mapping to MICO. For Java mapping the Java ORBacus 4 package [7] is used.

2.2 Client-Server Software Components

Fig. 1 illustrates the conceptual design of the client-server software components. The C++ based server components run exclusively on the "Model Server". The Tcl/Tk and Java based clients reside on the consoles. For a detailed description of the various server software components see [8].

3 COMMISSIONING EXPERIENCE

3.1 System Software Components

The chosen operating system Linux has demonstrated its reliability ("Model Server" uptimes of several months), flexibility (shell script based start/stop mechanism for about eighty CORBA based services) and performance (CPU intensive services like the TRACY [9] services).

MICO has proven to be a reliable "backbone" ORB. The extensive use of event driven supply of information (mainly BPM and magnet data) revealed some problems when using the Event Service. But they could be resolved by carefully balancing the supply and demand of event data.

0-7803-7191-7/01/$10.00 ©2001 IEEE.

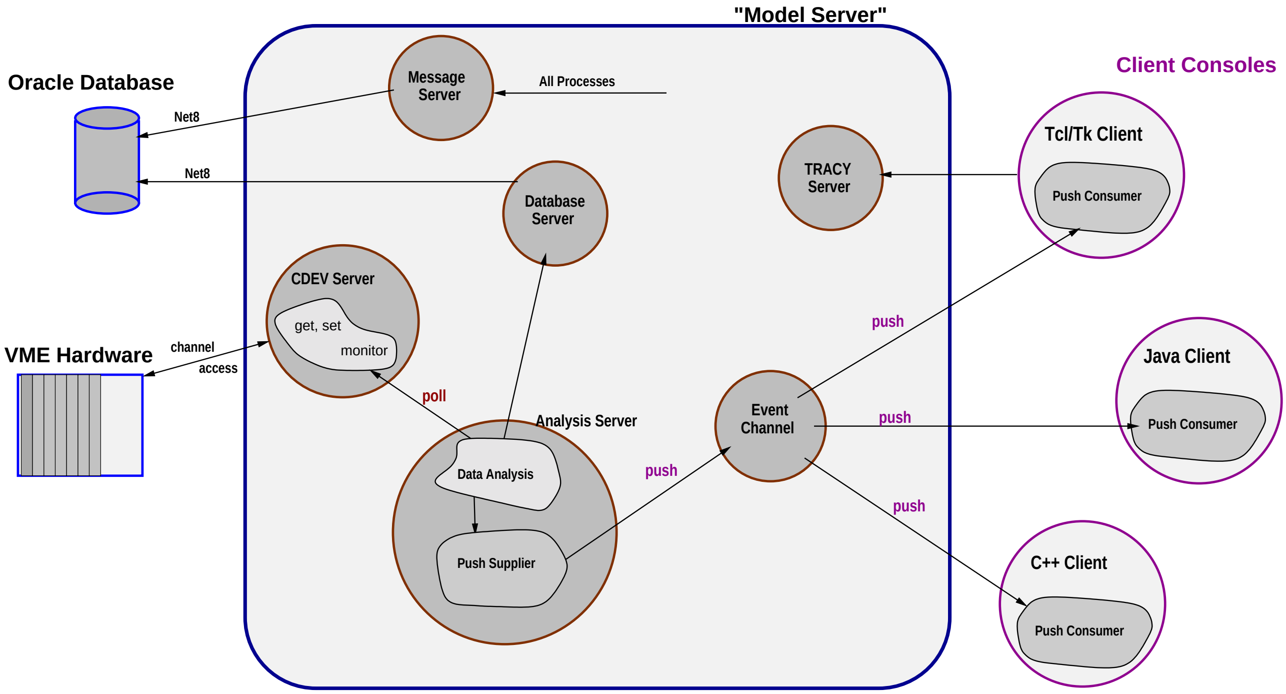

Figure 1: Conceptual design of the client-server software components.

3.2 *Client-Server Components*

- The **CDEV Server** gives access to the EPICS-based control system [10] through the Common DEVice (CDEV) library [11]. The server has been subject to various changes. Especially subtle bugs like tiny memory leaks were difficult to find. Dedicated CDEV Servers have been introduced whenever the hardware demanded a special service (e.g. digital BPM System [12]).
- The **Analysis Server** retrieves “raw” data from other servers like the CDEV Servers, analyzes them and distributes the result to clients through a CORBA event channel. Analysis Servers for BPM, corrector and quadrupole data have been introduced. This type of server is also well suited for temporary solutions like the automatic gain control of the digital BPM system which is supposed to be implemented low level later on.
- The **Database Server** provides access to the Oracle instances. Up to now it is only employed for accessing static accelerator data. The database access through the Oracle Template Library OTL [13] and the (OCI8) Oracle Call Interface in conjunction with the use of Binary Large OBjects (BLOB) predestinates the server for fast “online” data storage.
- The **Message Server** employs the UNIX syslog message logging facility. It profits directly from the reliability of standard UNIX services.
- The **TRACY Servers** provide access to the TRACY based models of transferlines, booster and storage ring. These servers have been subject to frequent changes as the machine model evolved. They have been developed in close relation to specific client applications which have no “model” knowledge by themselves like the orbit correction or the phase space display client.

Various client applications have been introduced during one year of booster and storage ring commissioning addressing the following tasks:

- **Transferline Optimization:** The linac-booster and booster-ring transferlines is modelled online. Corresponding twiss parameters, beam envelopes and orbit data are displayed. After optimization both transferlines show transmission rates close to 100% at booster and storage ring injection efficiencies of 60% and up to 100%.
- **Orbit Measurement:** “Bare orbits” (correctors switched off) of x_{rms}=1.8 mm and y_{rms}=0.7 mm have been observed in the storage ring which is consistent with rms quadrupole displacement errors of 30 μm. The vertical dispersion after orbit correction has been measured to be only ≈5 mm rms (see Fig. 2).
- **Global orbit correction:** An SVD based scheme is employed for booster and storage ring orbit correction. In case of the storage ring residual orbits of $x_{rms} = 7\mu$m and $y_{rms} = 1\mu$m could be obtained and kept for several hours which demonstrates the capabilities of the digital BPM hardware and indicates a

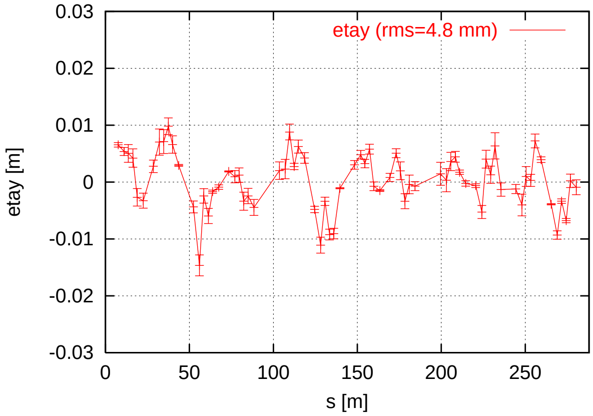

Figure 2: Vertical dispersion in the storage ring after orbit correction.

good long term stability. Horizontal mean orbit correction is provided by means of a central frequency correction.

- **Corrector/BPM response matrix measurement:** Response matrices have been measured for the storage ring and compared to the model. Good agreement has been found but an SVD based fit of the model to the orbit data still needs to be carried out. Based on turn by turn BPM (tune) data an SVD based beta beat correction on average betas derived from individual quadrupole variations has been performed reducing the horizontal and vertical beta beat to 5% and 2.8% [2].
- **Beam Based Alignment (BBA):** BBA has just been started. First measurement have been performed using a "static" BBA where the focussing of individual quadrupoles is changed by a fixed increment. "Dynamic" BBA (modulation of the focussing at a few Hz) will be carried out at a later stage.
- **Local orbit control through closed orbit bumps:** Symmetric and asymmetric 4-bumps have been used to steer the beam in the vicinity of the already installed in vacuum undulator.
- **Aperture scan utilizing closed bumps (booster at injection energy):** Interleaved 3-bumps have been scanned in case of the booster at injection energy in order to center the beam within the aperture of the beam pipe.
- **Display of turn by turn BPM data (e.g. phase space display):** The turn by turn capability of the BPM system allows to display "live" phase space plots. The display of turn by turn rms orbit data has been used to reduce the residual orbit oscillation outside of the storage ring injection kicker bump to $x_{rms} = 150\mu$m in the horizontal plane and effectively no orbit oscillation in the vertical plane within the precision of the orbit measurement.
- **Tune/Chromaticity measurement:** Turn by turn data are analyzed by means of a Fast Fourier Transformation. Peak identification of the spectrum reveals the tunes.
- **Lifetime measurement:** Lifetimes are fitted to the Parametric Current Transformer (PCT) readings. At the design current of 400 mA Touschek limited lifetimes of 3.5 hours have been observed.

4 CONCLUSION

CORBA based beam dynamics applications have been successfully used for the commissioning of the SLS booster and storage ring. The chosen framework proved to be flexible and reliable at an early stage of the commissioning allowing to concentrate on the "real" problems during commissioning. The natural integration of TRACY based online models within this framework allowed for a fast assimilation of these models. One of the next important tasks will be the implementation of a slow (2 Hz) and fast (4 KHz) global orbit feedback.

5 REFERENCES

[1] M. Böge *et al.*, "The Swiss Light Source Accelerator Complex: An Overview", EPAC'98, Stockholm, June 1998.

[2] A. Streun *et al.*, "Commissioning of the Swiss Light Source", Contribution to this Conference.

[3] Object Management Group (OMG), "The Common Object Request Broker: Architecture and Specification, Revision 2.2", February 1998.

[4] M. Böge, J. Chrin, "A CORBA Based Client-Server Model for Beam Dynamics Applications at the SLS", ICALEPCS'99, Trieste, October 1999.

[5] A. Puder, K. Römer, "MICO, An Open Source CORBA Implementation", 3rd Edition, Pub: dpunkt.verlag, Heidelberg, December 1999.

[6] F. Pilhofer, "Combat, CORBA scripting with Tcl", http: //www.informatik.uni-frankfurt.de/~fp/Tcl/tclmico/.

[7] http://www.ooc.com/.

[8] M. Böge *et al.*, "Development of Beam Dynamics Applications within a CORBA Framework at the SLS", EPAC 2000, Vienna, June 2000.

[9] J. Bengtsson, "TRACY-2 User's Manual", SLS Internal Document, February 1997; M. Böge, "Update on TRACY-2 Documentation", SLS Internal Note, SLS-TME-TA-1999-0002, June 1999.

[10] S. Hunt *et al.*, "Control and Data Acquisition System of the Swiss Light Source", ICALEPCS'99, Trieste, October 1999.

[11] J. Chen *et al.*, "CDEV: An Object-Oriented Class Library for Developing Device Control Applications", ICALEPCS'95, Chicago, November 1995.

[12] V. Schlott *et al.*, "Commissioning of the SLS Digital BPM System", Contribution to this Conference.

[13] S. Kuchin, "Oracle and Odbc Template Library Programmer's Guide", http://home.sprynet.com/~skuchin/.