OPEN ARCHITECTURE SOFTWARE INTEGRATION SYSTEM (OASIS) FOR THE PARTICLE BEAM OPTICS LABORATORY (PBO LAB)

G. H. Gillespie and B. W. Hill

G. H. Gillespie Associates, Inc., P. O. Box 2961, Del Mar, CA 92014, USA

Abstract

The Particle Beam Optics Laboratory, or PBO Lab™, is a suite of software applications developed to support beamline design, accelerator operations, and personnel training. The software provides an intuitive and easy-to-use graphic user interface (GUI) that works with a variety of particle optics programs. The PBO Lab GUI is largely responsible for the popularity of this software suite, which is now used at more than ninety institutions throughout the world. Although PBO Lab greatly improves the human-machine interface for several popular optics programs, it has historically required a significant effort to incorporate additional optics programs into the software suite. The Open Architecture Software Integration System, or OASIS™, provides an innovative framework that allows users to readily integrate their own optics programs into PBO Lab. This paper provides an overview of the OASIS framework and describes some of the work aimed at creating new PBO Lab Modules using OASIS.

INTRODUCTION

The Particle Beam Optics Laboratory was introduced a decade ago at the 1997 Particle Accelerator Conference [1]. PBO Lab has evolved considerably since that time and now provides a set of software application modules developed to support beamline design, personnel training, and accelerator operations [2]. PBO Lab provides a common graphic user interface (GUI) for constructing and editing iconic object-based computer models of accelerators and beamlines. The broad range of applications that may be addressed, together with the ease-of-use provided by the intuitive GUI, has made PBO Lab a popular package used throughout the world [3].

PBO Lab has had a variety of optics program modules available for some time [2-5]. Each optics program module incorporates a specific particle optics modeling or simulation program. Other modules are also available, including an Optimization Module. The Optimization Module works in conjunction with the optics program modules and can be used to address problems that are more complex than those that can be treated by the optics programs alone [6]. The common GUI for optics programs in PBO Lab simplifies the task of carrying out calculations for a given beamline. PBO Lab assures that the input files for the optics programs represent the same beamline, and PBO Lab will automatically display graphical results that may otherwise require post-processors.

The suite of optics programs historically available as PBO Lab Modules meets the needs of many users. Nevertheless there are a great many other particle beam optics programs used in the accelerator community [7]. A common request from PBO Lab users has been to integrate one or more of those programs into the PBO Lab GUI environment. Unfortunately the development of a new module typically required a C++ programming effort that could not be justified by the end user's anticipated usage of the module.

A new approach to developing a PBO Lab module was needed. The approach would have to provide a method for developing any one module with only a modest effort - an effort that would be commensurate with the relatively small utilization that a single module might receive. The concept for the Open Architecture Software Integration System was based upon these considerations. A proposal was submitted to the U. S. Department of Energy Small Business Innovation Research (SBIR) Program in 2004. The feasibility of the OASIS concept was demonstrated during the SBIR Phase I work and the project was selected for a Phase II development effort. This paper summarizes the OASIS framework and the progress achieved to date in developing new PBO Lab modules with the framework.

THE OASIS FRAMEWORK

The goal of the development effort was to provide the user with a software framework that could be utilized to define and *implement* the specifications for a new PBO Lab module *without* that user needing to write any (C++) code for the GUI part of the module. The physics (beam optics) program would be linked as a run time library. Ideally the physics program would also require no new code other than that which might be needed to make the program into a library rather than a standalone executable. To achieve this goal the OASIS framework is built upon the Multi-Platform Shell for Particle Accelerator Related Codes [8] that was developed to create the PBO Lab application suite.

OASIS incorporates a number of extensions to the object model [8] that underlies the PBO Lab software. Prominent among these extensions are a syntax parser and interpreter. The syntax parser and interpreter alleviate the need for any new C++ code. The OASIS Module Builder (described further below) is used to develop the target program's input file format and syntax for all of the optics

05 Beam Dynamics and Electromagnetic Fields　　　　　　D05 Code Developments and Simulation Techniques

elements that the target program supports. This process results in a custom PBO Lab Application Module called an OASIS Module, or simply OModule. The syntax parser and interpreter then operate dynamically at run time, via the OASIS Support Module (also discussed further below), on the PBO Lab object model of a beamline to create the input file to be read and acted upon by the target optics program. Once an OModule has been created using the OASIS Module Builder, the resulting module can be distributed for use with any PBO Lab software containing the OASIS Support Module. A PBO Lab user interacts with the new module in the same familiar way that he or she interacts with any other module. Little or no training is required to begin using the target optics program that is incorporated as a library into the new module.

The OASIS Module Builder

New modules are constructed with the OASIS Module Builder, or simply OModule Builder. The OModule Builder has a graphic interface that is very similar to that of PBO Lab. Module developers experienced with PBO Lab will find the environment familiar, and the procedures used to define a new module will be similar to various procedures used in PBO Lab. For example, many OModule Builder procedures begin by selecting an item from the OASIS Commands menu. This OModule Builder menu appears alongside entries for other modules in the PBO Lab Commands menu. Figure 1 shows the OASIS Commands menu.

When an item on the OASIS Commands menu is selected a window opens that provides the interface for specifying module details corresponding to that item. For example, the OASIS Module Specification item opens a window that provides entry fields used to define the "top level" OModule specifications. These top level specs include the module name (e.g. PARMILA), the physics program library (e.g. PARMILA.dll), library entry point (PARMILA_INTERFACE), and other information such as module version number, author, etc., that the module developer wishes to include. Certain fields have required entries (such as the module name) while others are optional (such as the module author).

Several of the windows associated with entries on the OASIS Commands menu have links to open additional windows, depending upon the complexity and detail needed to define the associated specifications. One example is the window that opens for the Module I/O File Specification. For graphics files (files that contain data that can be graphed or plotted) another window is opened in order to spec the type of graphic(s) to be displayed (e.g. line plots, scatter plots, histograms) and the format of the data in the associated file created by the physics program.

Figure 1: The Commands menu for the pre-release version of the OASIS Module Builder with a brief summary of each block's primary function.

There are a fair number of detailed specifications that may be required in order to fully define the requirements for writing an input file for a physics program, where all data is extracted from the PBO Lab object model. The items on the Commands menu encapsulate several of them, but not all items are used for each module definition. For example, the Lines and Sublines Specification item is used to define the structure of hierarchical beamline representations, when the associated physics program (e.g DIMAD) uses such a representation. Many programs (e.g. TRACE 3-D) do not use a hierarchical representation and there is no Lines and Sublines Specification to define for those programs.

Most of the detailed specifications for describing individual optics elements are defined using familiar PBO Lab Pieces [1]. Piece Windows (e.g. for a quadrupole) are used to select individual parameters (e.g. quadrupole length, strength, etc.) for use with the module. The OASIS Module Builder implementations of the Piece Windows contain links to other windows used to define the detailed syntax specifying the format for writing those parameters to the module's input file.

It is only possible to give a brief overview the OASIS Module Builder in a paper of this type. A detailed user manual is in preparation, and drafts of that manual [9] have been distributed to PBO Lab users assisting us in the development of modules for specific optics programs.

Once a new module has been defined, the module can be exported from the OASIS Module Builder as a PBO Lab module definition file (extension .omod). The module definition file encapsulates all of the definitions and specifications in a format that corresponds to the persistent object model used by PBO Lab. The OASIS Support Module is used to instantiate OModules.

The OASIS Support Module

The OASIS Support Module implements a generic design pattern for PBO Lab Application Modules. The Support Module parses and interprets the OModule specifications (.omod files) to instantiate all OModules at runtime. Each OModule encapsulates into the design pattern the details of a particular particle beam optics program. Multiple OModules may be instantiated so that multiple beam optics programs may be utilized together.

The OASIS Support Module is integrated into the PBO Lab software as an optional module. If no OModules are present, the OASIS Support Module is dormant and PBO Lab operates normally as configured for that user. If OModules are present then they appear in the PBO Lab GUI similar to the standard Application Modules that have been available for some time. The new OModules work side-by-side with existing PBO Lab Modules with an implementation that is seamless to the user. OModules allow users to leverage for their own optics programs the advanced GUI, tools, and object model of the PBO Lab software.

OModules Under Development

Several OModules are currently under active development. A number of different beam optics programs were selected for this development. The particular programs selected span a spectrum of program types. The diverse program types helped to refine the module design pattern to be as generic as possible.

Research partners are assisting us in testing the OModules. OModules that have received considerable attention to date are for the beam optics programs PARMILA and DIMAD. A prototype PARMILA OModule has also been used in a recent USPAS course (http://uspas.fnal.gov/programs/MSU/low-beta.html).

SUMMARY

An open architecture framework is being developed that will allow users of the PBO Lab software to create modules for their own beam optics programs. The framework implements several extensions to the existing PBO Lab software object model. Two new additions to

the PBO Lab architecture are the OASIS Module Builder and the OASIS Support Module. The OASIS Module Builder provides a PBO Lab type of GUI which is used to develop custom Application Modules called OModules. Several OModules are being developed using the OASIS Module Builder. The OASIS Support Module is used to instantiate OModules in PBO Lab. Testing of these and other modules is being carried out both internally and in collaboration with partners using PBO Lab with the OASIS Support Module.

ACKNOWLEGEMENT

REFERENCES

[1] G. H. Gillespie, B. W. Hill, H. Martono, J. M. Moore, N. A. Brown, M. C. Lampel and R. C. Babcock, "The Particle Beam Optics Interactive Computer Laboratory for Personal Computers and Workstations," Proc. 1997 IEEE Part. Accel. Conf., Vol. 2, 2562-2564 (1997).

[2] G. H. Gillespie, B. W. Hill, H. Martono and J. M. Moore, "Recent Advances in Software for Beamline Design, Accelerator Operations and Personnel Training," Nucl. Instr. Meth. B 161/3, 1168-1173 (2000).

[3] PBO Lab is available from AccelSoft Inc., San Diego, California, http://www.ghga.com/accelsoft.

[4] G. H. Gillespie, B. W. Hill, H. Martono, J. M. Moore and A. J. Dragt, "Using MARYLIE with the Particle Beam Optics Laboratory," Proc. 1999 IEEE Part. Accel. Conf., Vol. 3, 1551-1553 (1999).

[5] B. W. Hill, G. H. Gillespie and J. M. Moore, "Using TRACE 3-D in the Particle Beam Optics Laboratory," XX Inter. Linac Conf. Proc., SLAC-R-561, Vol. 2, 824-826 (2000).

[6] G. H. Gillespie, B. W. Hill, J. M. Moore and M. D. Woodley, "Some New Approaches to Solving Constrained Transfer Line Matching Problems," XX Inter. Linac Conf. Proc., SLAC-R-561, Vol. 1, 306-308 (2000).

[7] The Los Alamos Accelerator Code Group, "Computer Codes for Particle Accelerator Design and Analysis: A Compendium" LA-UR-90-1766, 290 pp (1990).

[8] B. W. Hill, H. Martono, J. M. Moore and J. S. Gillespie, "An Application Framework and Intelligent Graphic User Interface for Multiple Accelerator Codes," Proc. 1998 Intern. Comp. Accel. Phys. Conf., SLAC-R-580, 92-96 (1998).

[9] "OASIS 0.8 (Open Architecture Software Integration System - SBIR Phase II Development Version) User Manual: Building and Using Application Modules for the Particle Beam Optics Laboratory (PBO Lab)," 126 pp (2007).