THE OPEN ARCHITECTURE SOFTWARE INTEGRATION SYSTEM (OASIS) FOR CREATING PBO LAB MODULES

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

Abstract

A specialized software package has been developed that enables the rapid implementation of custom beam optics modules that run in the Particle Beam Optics Laboratory (PBO Lab™). PBO Lab is a commercially available software application that supports a suite of accelerator codes for design, operations, and personnel education. The intuitive and easy-to-use graphic user interface (GUI) is largely responsible for the popularity of PBO Lab. The Open Architecture Software Integration System, or OASIS, builds upon the capability of PBO Lab to host a suite of different codes. OASIS provides an innovative framework that allows users to readily create new PBO Lab modules without writing or compiling any source code. OASIS has been used to develop several new modules for PBO Lab. This paper presents a summary of the OASIS framework and describes some of the features used in creating a new PBO Lab module for one example optics code.

INTRODUCTION

PBO Lab has been used to support beamline design, personnel training, and accelerator operations for over a decade and delivers a suite of Application Modules that implement a variety of trusted optics codes [1]. The PBO Lab GUI provides a common graphic user interface for creating and editing iconic object-based computer models of accelerators and beamlines.

OASIS leverages the mature software framework utilized by PBO Lab [2], and extends its capabilities to provide users with a familiar environment from which to create their own PBO Lab Application Modules [3]. The PBO Lab Beamline Object Model [4] is the underlying persistent representation that allows multiple optics codes with varying input/output requirements to utilize a single beamline model description. Multiple optics codes run side by side in PBO Lab's sophisticated drag-and-drop graphic user interface. PBO Lab generates the native input and processes the native output, for the different optics codes, from a single beamline model.

THE OASIS FRAMEWORK

The OASIS framework is a collection of abstract and concrete classes, and their interfaces which comprise a generic subsystem used to implement Application Modules in PBO Lab. It provides a reusable, generic OASIS Module, or "OModule," that encapsulates the user interface and data interface functionality required for running in the multi-module PBO Lab environment. Figure 1 illustrates the abstract layer diagram for the PBO Lab/OASIS application framework. User developed Application Modules are implemented dynamically in PBO Lab with this generic reusable Application Module (OModule) design pattern.

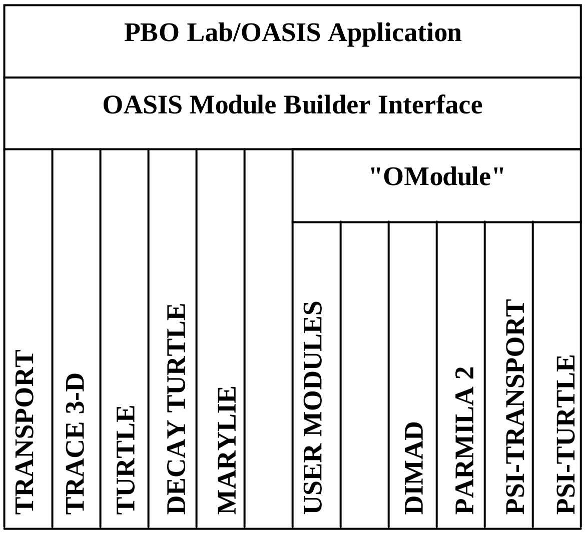


Figure 1: Layer diagram for PBO Lab Framework and Open Architecture Software Integration System.

An OModule has both a user interface component and a computational component. The user interface component is created in the OASIS Module Builder Interface which is an extension to the drag-and-drop PBO Lab GUI. The computational component may be a library (.dll) or executable (.exe) application. No source code needs to be compiled or linked to use the Application Module in PBO Lab.

OASIS MODULE BUILDER

The OASIS Module Builder Interface, used to create PBO Lab Application Modules, is an intuitive extension of the standard PBO Lab interface, providing users with a familiar environment to create custom Application Modules that run seamlessly in the PBO Lab GUI without writing or compiling any source code. The OASIS Module Builder is itself a very specialized PBO Lab Module: it is a Module for the specification of other Application Modules. The OASIS Module Builder provides an interactive environment to incrementally create and test Application Modules in the PBO Lab GUI. Detailed specifications to fully define the requirements for the generation of a native input file for an optics code are accomplished through the familiar PBO Lab GUI, which has been extended to support the OASIS Module Builder Interface.

Application Modules are exported from the OASIS Module Builder for use in PBO Lab using a PBO Lab module definition (.omod) file. Figure 2 illustrates the process. The module definition file utilizes an enhanced version of the PBO Lab persistent Beamline Object Model representation [4]. The object model has been expanded to include specifications that define the native input/output requirements for a given optics code, and to include a variety of other specifications and constraints that encapsulate the operations and commands of the target optics code.

Iconic representations (Piece Components) of physical beamline elements and non-physical beamline operations, in the Module Builder Interface, are used to access and specify most input requirements for a target optics code.

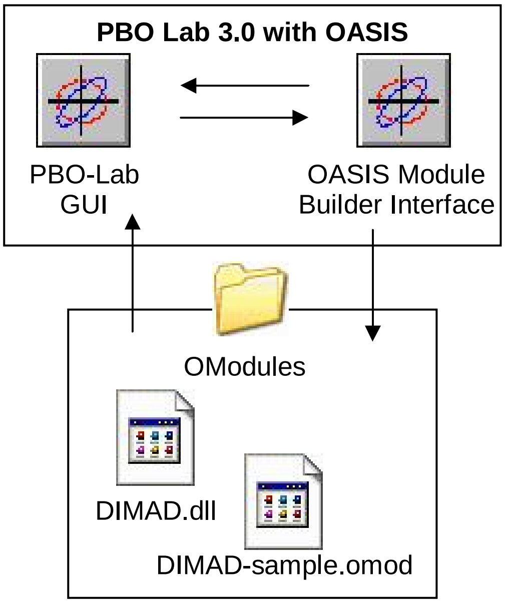


Figure 2: Application Modules can be interactively developed and executed in the Module Builder Interface and exported for use directly in the PBO Lab GUI as (.omod) OModule definition files.

MODULE BUILDER EXAMPLE

Piece Components are categorized into three types, Optics Pieces, Parameter Pieces and Command Pieces. Optics Pieces are used to represent physical beamline elements and include a variety of standard Pieces available in the PBO Lab Palette Bar [5]. Customizable Optics Pieces can be used to add additional optics elements. The Parameter Piece type provides a customizable generic interface and is typically used to support parameters not associated with any particular physical optics element. Command Pieces are used to implement Application Module commands or operations supported by the target optics code. Command Pieces provide a versatile method of encapsulating the wide variety of commands that can be specified for a given optics code. Command Pieces do not represent physical beamline elements, although they can be located at specific locations in the beamline model, where they are evaluated sequentially to produce the appropriate command instructions in the native input file for the given optics code.

Each optics code has its own unique “commands” or “operators” that encapsulate the computational capability for that code. This code-specific functionality is implemented in an Application Module using a set of Command Pieces. An example of such a Command Piece, created for a DIMAD Application Module, is illustrated here.

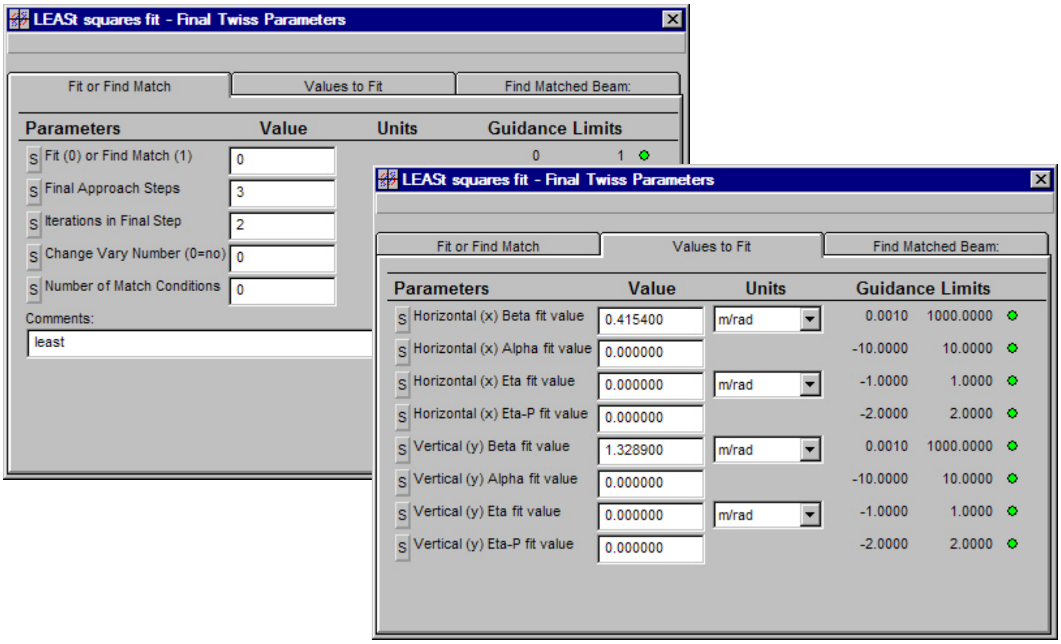
The DIMAD program [6] was developed to study particle motion in both beamlines and circular machines. The code uses a matrix formalism through second order. It is perhaps most commonly used at facilities operating electron accelerator systems.

There are several Command Pieces defined for the DIMAD Application Module. For illustrative purposes, the implementation of DIMAD's Least Squares Fit operation will be described here. (The Least Squares Fit operation is described in Section 2.15 of reference [6].)

The DIMAD Application Module “Least Squares Fit…” Command Piece implements a variety of native instructions required to direct the DIMAD computation engine to perform a particular fitting operation [6]. The definition of the “Least Squares Fit…” Command Piece in the OASIS Module Builder Interface involves both the specification of the required input parameters and the syntax used to generate instructions for the DIMAD native input file.

A generic Command Piece serves as a template and is initially created on the Work Space [5] of the Module Builder Doc Window using a menu command that prompts the user to name the new Command Piece. The named piece can then be dragged to the Model Space [5] where double-clicking the piece icon will open its Command Piece Specification Window. From there, additional windows are accessed that provide for the definition of input parameters and the specification of the native syntax required for the command to be processed by the target optics code (i.e. DIMAD). Once the Command Piece is defined, it will be available for use with the DIMAD Application Module in PBO Lab. Figure 3 shows two of the input parameter tab pages defined in this way for the “Least Squares Fit…” Command Piece.

The “Least Squares Fit…” Command Piece is added to a PBO Lab beamline to direct the DIMAD computation engine to perform the prescribed fitting operation. From the DIMAD Command Menu, the “Write and Run DiMadInput” command is used to generate the native DIMAD input, execute the DIMAD computation engine, and open the resulting output file.



Parameters	Value	Units	Guidance Limits	
Fit (0) or Find Match (1)	0		0	1
Final Approach Steps	3			
Iterations in Final Step	2			
Change Vary Number (0=no)	0			
Number of Match Conditions	0			

Comments:
least

Parameters	Value	Units	Guidance Limits	
Horizontal (x) Beta fit value	0.415400	m/rad	0.0010	1000.0000
Horizontal (x) Alpha fit value	0.000000		-10.0000	10.0000
Horizontal (x) Eta fit value	0.000000	m/rad	-1.0000	1.0000
Horizontal (x) Eta-P fit value	0.000000		-2.0000	2.0000
Vertical (y) Beta fit value	1.328900	m/rad	0.0010	1000.0000
Vertical (y) Alpha fit value	0.000000		-10.0000	10.0000
Vertical (y) Eta fit value	0.000000	m/rad	-1.0000	1.0000
Vertical (y) Eta-P fit value	0.000000		-2.0000	2.0000

Figure 3: Input parameter tab pages for the “LEASt Squares Fit…” Command Piece window.

The first four letters of the Command Piece name, "LEASt squares fit ..." are intentionally capitalized: those letters represent the DIMAD keyword for this operation. However, users need not know this keyword, nor learn any of the other input file syntax and format for DIMAD's least squares fitting operation. The Command Piece Window (Figure 3) provides text descriptions of all parameters. The OASIS Module Builder syntax specification for this Command Piece provides PBO Lab with the necessary instructions needed to write the parameters in the correct format, and in the correct location, to the DIMAD input file. Users simply set the fitting goals using the parameters of the Command Piece Window illustrated in Figure 3.

The least squares fitting computation is just one example of available operations implemented for the DIMAD Application Module. A variety of other calculations are also implemented using other DIMAD Command Pieces.

SUMMARY

A specialized software package has been developed called the Open Architecture Software Integration System, or OASIS, that can be used to create custom Modules for the Particle Beam Optics Laboratory (PBO Lab). OASIS builds upon the capability of PBO Lab to host a suite of different codes, and provides an innovative framework that enables users to rapidly implement custom beam optics modules. The custom modules will run in PBO Lab without writing or compiling any source code.

ACKNOWLEDGEMENT

Parts of this work were supported by the U. S. Department of Energy Small Business Innovative Research program under grant number DE-FG02-04ER83961.

REFERENCES

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

[2] 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).

[3] G. H. Gillespie and B. W. Hill, "Open Architecture Software Integration System (OASIS) for the Particle Beam Optics Laboratory (PBO Lab)," Proc. 2007 Particle Accelerator Conf., paper THPAS037 (2007).

[4] B. W. Hill, H. Martono and J. S. Gillespie "An Object Model for Beamline Descriptions," AIP Conf. Proc. 391, 361-365 (1996).

[5] 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).

[6] R. V. Servranckx, K. L. Brown, L. Schachinger, D. Douglas and P. G. Tenenbaum, "Users Guide to the Program DIMAD," SLAC Report (2004).