

SANDIA REPORT

SAND2010-xxxx

Unlimited Release

Printed March 2006

Supersedes SAND1901-0001

Dated January 1901

Optika: A GUI Framework for Parametrized Applications (Report style, strict)

Kurtis L. Nusbaum

Prepared by

Sandia National Laboratories

Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Approved for public release; further dissemination unlimited.



Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: <http://www.osti.gov/bridge>

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: <http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online>



SAND2010-xxxx
Unlimited Release
Printed July 2010
Reprinted March 2006

Supersedes SAND1901-0001
dated January 1901

Optika: A GUI Framework for Parametrized Applications (Report style, strict)

Kurtis L. Nusbaum
Scalable Algorithms
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-9999
klnusba@sandia.gov

Abstract

In the field of scientific computing there are many specialized programs designed for specific applications in areas like biology, chemistry, and physics. These applications are often very powerful and extraordinarily useful in their respective domains. However, many suffer from a common problem: a poor user interface. Many of these programs are homegrown, and the concern of the designer was not ease of use but rather functionality. The purpose of Optika is to address this problem and provide a simple, viable solution. Using only a list of parameters passed to it, Optika can dynamically generate a GUI. This allows the user to specify parameters values in fashion that is much more intuitive than the traditional "input decks" used by many parameterized scientific applications. By leveraging the power of Optika, these scientific applications will become more accessible and thus allow their designers to reach a much wider audience while requiring minimal extra development effort.

Acknowledgment

Thanks to Dr. Mike Heroux and Jim Willenbring. Their mentoring has been crucial to the development of Optika. Also, many thanks to the entire Trilinos Community in which Optika has found a welcoming home.

The format of this report is based on information found in [11].

Contents

Nomenclature	8
1 Introduction	9
2 Development of the Optika package	11
Initial Planning	11
Early Development	13
Heavy development	13
References	15

List of Figures

2.1	GUI Layout	12
2.2	Some of the various widgets used for editing data	12

List of Tables

Nomenclature

Dependency A relationship ship between to parameters in which the state or value of one parameter depends on the state or value of another.

Dependee The parameter upon which another parameter state or value dependes.

Dependent A parameter whose state or value is determined by another parameter.

Parameter An input needed for a program.

Parameter List A list of parameters and other parameter lists.

RCP Refernce counted pointer. RCPs refered to in this document reference the RCP class located in the Teuchos package of Trilinos.

Sublist A parameter list contained within another parameter list.

Widget A GUI element, usually used to obtain user input.

Chapter 1

Introduction

This report is comprised of two main sections. The first section discusses the development of Optika. Design choices and problems that arrived during development are discussed in this section. The second section is intended to be a manual on how to use Optika. If the reader desires simple to learn how to use Optika, it is suggested he/she skip to the second section.

This page intentionally left blank.

(Well, it's not blank anymore...)

Chapter 2

Development of the Optika package

This section of the report discusses the development of the Optika package.

Initial Planning

In Fall of 2008, Dr. Mike Heroux identified a need for the Trilinos framework to include some sort of GUI package. Dr. Heroux wanted to give users of the framework the ability to easily generate GUIs for their programs, while still providing a good experience for the end user. Based on previous GUI work done for the Tramonto project, a few initial problems were identified:

- How would the GUI be laid out?
- Different types of parameters require different methods of input. How would we decide how we would obtain input for a particular parameter?
- What framework would we use to build the GUI?
- How would the application developer specify parameters for the GUI to obtain?
- How would the application developer specify dependencies between parameters. This was a crucial problem/needed feature that was identified in development of the Tramonto GUI.

After some deliberation, the following initial solutions were decided upon:

- The GUI would be laid out in a hierarchical fashion as shown in Figure 2.1. Parameters organized into lists and sublists. This would allow for a clear organization of the parameters as well as intrinsically demonstrate the relationships between them.
- It be required that all parameters specify their type and the following types would be accepted:
 - int
 - short

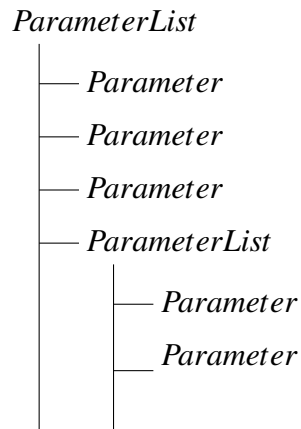


Figure 2.1. The hierarchical layout of the GUI

- float
- double
- string
- boolean
- arrays of int, short, double, and string

For number types, a spin box would be used as input. If the valid values for a string type were specified, a combo box would be used. Otherwise a line edit would be used. For booleans, a combo box would also be used. For arrays, a pop-up box containing numerous input widgets would be used. The widget type would be determined by the array type.

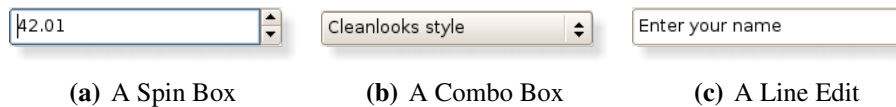


Figure 2.2. Some of the various widgets used for editing data

- QT was chosen as the GUI framework for several reasons:
 - It is cross-platform.
 - It is mature and has a well developed set of development tools.
 - It has a rich feature-set.
 - It has been used by Sandia in the past.
 - The Optika developer was familiar with it.
- Initially it was decided that the application developer would specify parameters via an XML file. A DTD would be created specifying the legal tags and name spaces.

- Dependencies would be handled through special tags in the DTD.

Early Development

The first several months of development were spent on creating and implementing the XML specification. The name of the XML specification went through several revisions but was eventually called Dependent Parameter Markup Language (DPML).

After several months of development it was realized that creating an entirely new way of specifying parameters might hinder its adoption. It pointed out that Trilinos actually had a `ParameterList` [10] class in the Teuchos package. The `ParameterList` seemed to be better than DPML for several reasons:

- It was already heavily adopted.
- It had the necessary hierarchical nature.
- It was serializable to and from XML.

For these reasons, DPML was scrapped in favor of using Teuchos's `ParameterLists`. Development moved forward with the goal of creating a GUI framework that, in addition to meeting all the challenges outlined above, would also be compatible with any existing program using Teuchos's `ParameterLists`.

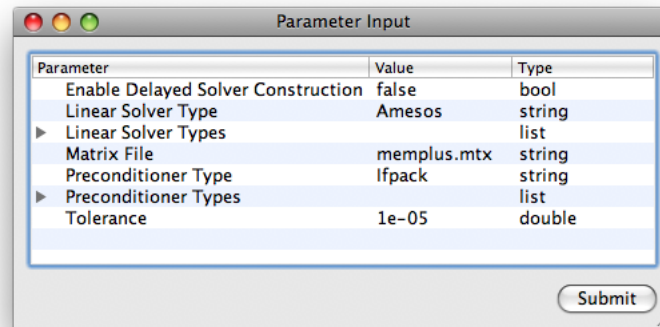
Heavy development

Starting in May 2009 a more heavy focus was put on development of the Trilinos GUI package. With the backend data-structure of the Teuchos's `ParameterList` already in place, attention was turned to the developing the actual GUI itself. A key technology provided by Qt was its Model/View framework [1]. Using the Model/View paradigm, a wrapper class named `TreeModel` [5] was created around the `ParameterList` class by subclassing the `QAbstractItemModel` [6].

However, in subclassing the `QAbstractItemModel` it was realized that the `ParameterList` class fell short in certain areas. At this point the main issue was that a given `ParameterEntry` [9] located within a `ParameterList` or a given sublist located within a `ParameterList` was not aware of its parent. This was an issue because Qt's Model/View framework requires items within a model to be aware of their parents. In order to circumvent this issue the `TreeItem` [4] class was created. Now instead of simply wrapping around a `ParameterList` class, the `TreeModel` created by giving it a `ParameterList`. It would then read in the `ParameterList` and create a structure of `TreeItems`. Each `TreeItem` then contained a pointer to its corresponding `ParameterEntry`.

Once the TreeModel and TreeItem class were complete an appropriate delegate to go between and View and the TreeModel was needed. A new class called Delegate [3] was created to fill this role by subclassing QItemDelegate [7]. As specified above, the delegate would return the appropriate editing widget based on its datatype.

With the model and delegate classes inplace, an appropriate view could be applied. At first a simple QTreeView [8] was applied to the model. The results was something like that in 2.3(a).



(a) A Tree View

References

- [1] Model/View Programming. <http://doc.trolltech.com/4.6/model-view-programming.html>.
- [2] Online Reference Documentation. <http://doc.trolltech.com>.
- [3] Optika::Delegate Class Reference. <http://trilinos.sandia.gov/packages/docs/10.3/packages/optika/DelegateClassReference.html>.
- [4] Optika::TreeWidgetItem Class Reference. <http://trilinos.sandia.gov/packages/docs/10.3/packages/optika/TreeWidgetItemClassReference.html>.
- [5] Optika::TreeModel Class Reference. <http://trilinos.sandia.gov/packages/docs/10.3/packages/optika/TreeModelClassReference.html>.
- [6] QAbstractItemModel Class Reference. <http://doc.trolltech.com/4.6/qabstractitemmodel.html>.
- [7] QAbstractItemModel Class Reference. <http://doc.trolltech.com/4.6/qitemdelegate.html>.
- [8] QTreeView Class Reference. <http://doc.trolltech.com/4.6/qtreeview.html>.
- [9] Teuchos::ParameterEntry Class Reference. <http://trilinos.sandia.gov/packages/docs/10.3/packages/teuchos/ParameterEntryClassReference.html>.
- [10] Teuchos::ParamterList Class Reference. <http://trilinos.sandia.gov/packages/docs/10.3/packages/teuchos/ParamterListClassReference.html>.
- [11] Tamara K. Locke. Guide to preparing SAND reports. Technical report SAND98-0730, Sandia National Laboratories, Albuquerque, New Mexico 87185 and Livermore, California 94550, May 1998.

DISTRIBUTION:

- 1 An Address
99 99th street NW
City, State
- 3 Some Address
and street
City, State
- 12 Another Address
On a street
City, State
U.S.A.

- 1 MS 1319 Rolf Riesen, 1423
- 1 MS 1110 Another One, 01400
- 1 M9999 Someone, 01234
- 1 MS 0899 Technical Library, 9536 (electronic)

Second Printing, (January 2006):

- 1 An Address
99 99th street NW
City, State
- 3 Some Address
and street
City, State
- 12 Another Address
On a street
City, State
U.S.A.

- 1 MS 1319 Rolf Riesen, 1423
- 1 M9999 Someone, 01234

Third Printing, (February 2006):

- 1 MS 1319 Rolf Riesen, 01423

Fourth Printing, (March 2006):

- 1 MS 1319 Rolf Riesen, 1423

This page intentionally left blank.

(Well, it's not blank anymore...)

