OptSuite – The Java Optical Measurement Suite

Alexander Bieber, Stephan Reichelt, and Hans Zappe

Laboratory for Micro-optics, Department of Microsystems Engineering – IMTEK,
University of Freiburg, Germany

alex@optsuite.org

A software platform for the control of experimental measurement setups is presented. The platform is built on the Eclipse RCP and makes use of its plug-in mechanism to build an extendable framework that can be adapted to a variety of tasks in an optical laboratory. OptSuite provides interfaces on all levels of programming or computer knowledge. It enables users to manage measurements by a graphical user interface or extend OptSuite's capabilities by JavaScript, Matlab scripts or Java classes. OptSuite's current set of plug-in's provides extensions of the framework for phase-shifting interferometry and time domain OCT. For this purpose, several reusable hardware interfaces and measurement algorithms were developed, including the control of FireWire cameras as intensity detectors, a framework for controlling motor stages, and USB-controlled piezo-driven device for phase-shifting.

1 Introduction

The control of experimental setups is often done individually in a variety of technologies or with expensive commercial software. This increases research costs and causes multiple implementations of similar tasks. The OptSuite[1] project aims to bundle common tasks arising from measurement applications in optical laboratories and to provide a framework that can be easily extended by users or programmers. Unlike commercial software, OptSuite is published as free software and under an Open Source license. It is written in the Java programming language, which has recently attracted a huge community and for which many libraries are freely available. OptSuite's focus of development has been on its general adaptability to different applications and on collaboration and sharing of its extensions.

2 Application Framework

The OptSuite platform is based on the Eclipse RCP[2] that provides a free application development framework. Although Eclipse is based on native libraries and adapts to the appearance and behavior of the underlying operating system, it is available for most common systems such as Windows, Linux or Mac. Besides basic UI components, like windows, buttons or tables, Eclipse provides a powerful plug-in architecture.

An Eclipse plug-in consists of its Java classes usually packed into a jar file and a descriptor (plugin.xml and MANIFEST.MF). The descriptor defines a plug-in's interconnections with others. Plug-ins can set dependencies on other plug-ins in order to access their exported classes. By providing an extension-point, a plug-in can declare an interface with which other plug-in's can interact, or extend the functionality provided by the declaring plug-in. The other plug-ins can access this functionality by declaring extensions to these points.

Furthermore, Eclipse plug-ins can be bundled into features that are used by Eclipse's update mechanism. They can be individually installed and updated from a remote update-site.

OptSuite is integrated into the Eclipse framework and defines extension-points specific to optical measurement. The next sections briefly describe the most important extension-points of OptSuite and their concepts.

3 Extension Concepts

OptSuite manages the different measurement tasks as so called measurement-routines. These routines consist of an arbitrary list of routine-steps and can be grouped into routine-sequences. One can consider a routine as the work flow of a real measurement setup that in most cases can be divided into three major parts: data-acquisition, data-analysis and data-visualization. All parts can be modeled as routine-steps, in which OptSuite provides special extension-points for common applications. One example is the measurement-tool concept of OptSuite. Measurement-tools are brought in by OptSuite plug-ins and provide access to hardware tools such as light sources, detectors or motor stages. The tools are made available to routine-steps upon data-acquisition.

For data-analysis, OptSuite simply passes the previously acquired data to registered data-analyzers, which then can either read the data for statistical or qualificatory purposes or directly manipulate it. Analyzers can add their results to so-called OptSuite reports. Reports have two aspects. One is the graphical result-presentation to the user and the second is the storage of these results. Here again a framework was developed and can be extended to support specialized UI representation or custom storage formats for the further use in external applications. Similar to the report framework, the extension-point "ioFilter" enables the registration of filters to store

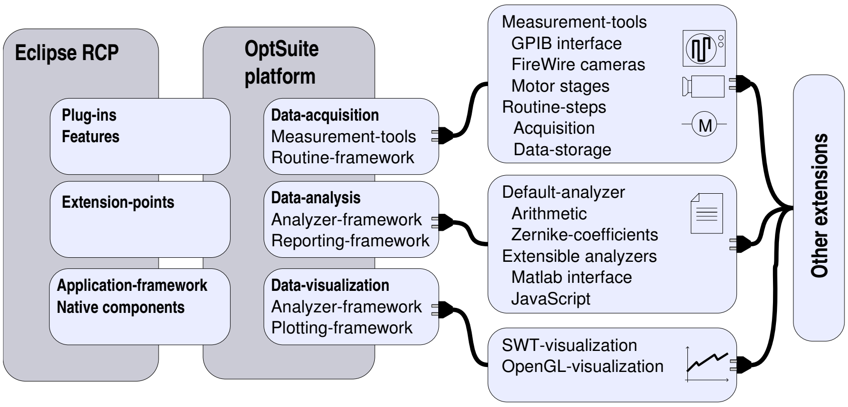

Fig. 1: OptSuite's architecture, which is based on the Eclipse RCP and provides extension-points to extenders of the platform.

the raw and analyzed measurement data to different file formats. Currently it supports the storage into binary and text format developed at IMTEK (*.opa, *.opb), CODE-V (*.int) and PGM format.

Data visualization is also done by special routine-steps. OptSuite incorporates plotting components in Java SWT technology that can be used to plot one- and two-dimensional data. These standard components, however, are again part of a framework in which OptSuite allows different plotting engines to visualize its data.

In addition to its build-in routine-steps or those provided by other plug-in developers, OptSuite allows its users to extend and customize the behavior of the application by so-called scripted routine-steps. These steps will execute a user-defined script written in either JavaScript or Matlab-script. Like Java routine-steps, the scripts have access to measurement-tools and the previously acquired data.

4 Available Extensions

OptSuite and its current extensions for phase-shifting interferometry (PSI)[3] and optical coherence tomography (OCT) provide features unique to these applications, but their development brought up several measurement-tools and routine-steps that can be used for general measurement or interferometrical applications.

A very abstract and flexible measurement-tool is the GPIB[4] interface, which provides a connection to GPIB devices and allows the execution of GPIB commands and queries. Together with a JavaScript routine-step, this can be used for a quick implementation of a device control. For positioning tasks, OptSuite provides a framework for the control of motor stages. Currently this can be used with the OWIS PCI card for linear stages.

Interferometry requires two-dimensional intensity detection that is modelled in OptSuites intensity-detector framework. Its standard extension can be used with digital cameras compliant to the IEEE 1394 standard (FireWire).

For data pre-processing, OptSuite includes a framework to mask the measurement data and mark its data points with attributes. Additionally, a graphical editor was developed to create masking areas for the acquired data.

Specialities of PSI are control of the reference phase-shift and unwrapping of the 2π-wrapped data that results from calculating the test phase data. Phase shifting is accomplished with a piezo transducer (P-239.00, Physik Instrumente). It is controlled via the USB-Port using an I/O controller (IO-warrior 40, Code Mercenaries) and D/A converter (MX7248, Maxim) to transform the USB output signal to the desired input signal of a high-voltage amplifier connected to the piezo transducer. For phase unwrapping, the Goldstein unwrapping algorithm was implemented in Java.

OptSuite's OCT extensions use the GPIB interface for data-acquisition and provide functions for the signal processing of OCT measurement data.

5 Conclusion

OptSuite provides a flexible framework for optical and general measurements. By publishing its source code, we are making it available to an interested optical community. With a growing number of routines and tools, OptSuite can shorten development time and limit duplicate implementations. We invite all interested parties to use OptSuite and to actively contribute to the project.

Acknowledgement

This work was partially financially supported by the Landesstiftung Baden-Württemberg gGmbH "Forschung Optische Technologien 2002".

References

1. http://www.imtek.de/micro-optics/optsuite
2. http://wiki.eclipse.org/index.php/Rich_Client_Platform
3. Stephan Reichelt and Hans Zappe, "Mach-Zehnder / Twyman-Green hybrid interferometer for micro-lens testing," DGaO-Proceedings, 2005. http://www.dgao-proceedings.de/download/106/106_a12.pdf.
4. http://en.wikipedia.org/wiki/GPIB