Workshop App: the Data Analysis Design Pattern of the Enabler Framework

Michael K. Kundmann

e-Metrikos, Pleasanton, USA

The Enabler framework project seeks to establish a unified context for exploring, integrating, and analyzing microscopy data sets captured with a broad range of techniques and instrumentation [1, 2]. Enabler uses object-oriented design techniques [3] to capture the concepts involved in thinking about, working with, and quantifying microscopy data in the form of fully functional, ready-to-use classes (software modules). A handful of classes form the essential basis of Enabler's data model: NumericBrick (a multi-dimensional array), PhysicalQuantity, PhysicalVector, PhysicalRange, PhysicalFrame, and PhysicalBrickDatum (a physically calibrated NumericBrick). Most microscopy data analyses involve measured data and output results that can be represented as some combination of these physical types.

Recent work on the framework has focused on developing a re-usable app architecture or design pattern [4] that can be deployed for the analysis of virtually any type of microscopy data set. The result is the Workshop App design pattern, which adds app and project controller base classes to Enabler's data model (fig. 1). To extract quantitative information from a particular type of multi-dimensional data, the analyst opens a workshop app tailored to that specific type of analysis. The view (UI) of the workshop app allows one to generate new or access existing project instances, each attached to a particular data set. The resulting project-datum controller network is an active mechanism taking one from measured data to desired outputs. Any analyst adjustment of data selections or project analysis parameters and options immediately triggers an update of output results. This workshop app architecture provides a high degree of workflow convenience and speed, as well as full traceability and transparency for peer review (fig. 2). Because Enabler workshop projects are connected, active objects, not listings of function calls or operator actions, they can embody and support arbitrarily complex data analysis paradigms, beyond simple processing pipelines. As fully supported Enabler framework objects, project controllers are auto-archiving, using Enabler's object archiving protocol to immediately save an analyst's adjustments and selections. The analyst's work is thus always secure and requires no invocation of a Save command. Such project archivability, capturing how an analysis was done, not just the final results, supports ready duplication of analyses, both for exploring different approaches to complex analyses, as well as for applying a carefully tailored analysis scheme to multiple, similar data sets. The Enabler framework has been implemented for the DM-scripting environment [5], but its concepts and archives can be deployed in and accessed from other software contexts. Enabler provides base classes for rapid project-based app development for both commercial (licensed) and community-serving (free) applications. The current release (version 3.4) is freely available to all interested parties upon request and detailed documentation about its app architecture and public class libraries will soon appear on the e-Metrikos website [6].

References:

[1] M. Kundmann, Microscopy and Microanalysis, 22 Suppl. 3, (2016) p. 290.
[2] M. Kundmann, 2018 M&M Meeting, late-breaking abstract and poster (un-published).

[3] G. Booch *et al*., "Object-Oriented Analysis and Design with Applications", 3rd ed. (Addison-Wesley, 2007) chs. 2 and 3.

[4] E. Gamma *et al*., "Design Patterns", (Addison-Wesley, 1994) p. 4.

[5] http://www.gatan.com/products/tem-analysis/gatan-microscopy-suite-software

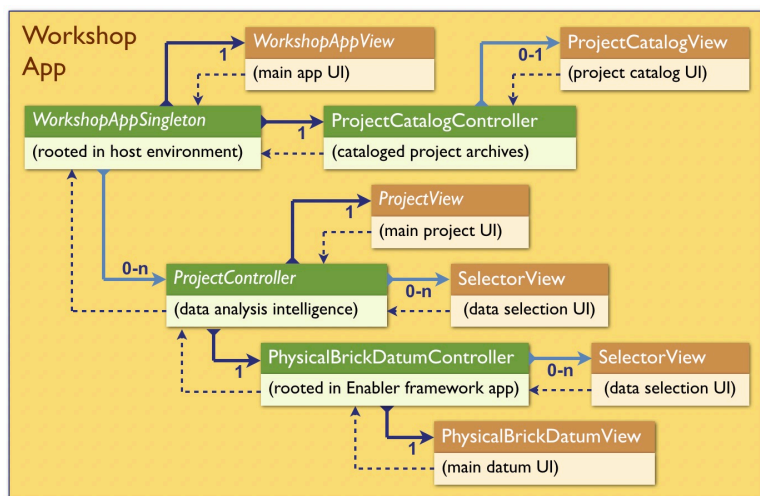[6] http://www.e-metrikos.com/enabler

Figure 1. Schematic UML class diagram of the Workshop App design pattern developed for the Enabler framework. Solid arrows represent parent-child instantiation relationships. Dotted arrows reflect back-communication pathways that can be used to keep all software modules apprised of analysis project status and analyst-triggered updates.

Figure 2. Snapshot of an active EELS spectrum image data set analysis using EELS Workshop, a typical realization of Enabler's Workshop App design pattern. All view and ROI actions are conveyed to the project via the underlying network of controller objects depicted in the class diagram.