APPLICATION BUILDER AND COMSOL SERVER™: A REVIEW

by **WILLIAM T. VETTERLING**

COMSOL Multiphysics® software version 5.0 introduces something new to the modeling enterprise: The Application Builder, and its counterpart, COMSOL Server™. The Application Builder allows the modeler to sweep away the detail-oriented tools that were used to produce a model and to fashion a more approachable application—an app—that is intuitive and easy to use.

The app may have a simple interface constructed with an assortment of buttons, lists, menus, graphics, and text to make its operation straightforward. It may be launched from within the COMSOL Multiphysics program. But, importantly, it may also be lifted entirely from that feature-rich environment with the help of a worldwide COMSOL Server license. In the latter case, the app may be run as a standalone application, or as a web resource within a browser.

There are many scenarios in which the Application Builder will find potential uses. With an app, modeling results may be presented in the form of live, real-time examples rather than static summaries in charts and graphs. Likewise, apps may be constructed for use in lectures or demonstrations. Companies may offer simulation apps demonstrating their product's performance, to be used in place of data sheets, or may produce licensable apps as products in their own right.

In brief, the Application Builder is a new line of communication between the professional modeling expertise of the model builder, and the science and engineering expertise of model users.

➔HOW TO BUILD AN APP

Creating an app is easy enough to try. I began with a functioning COMSOL Multiphysics model that I wanted to turn into an app. I chose to use a simple 3D model of the current distribution and heating of a multi-pixel thick-film thermal print head such as might be used in a fax machine (see Figure 1).

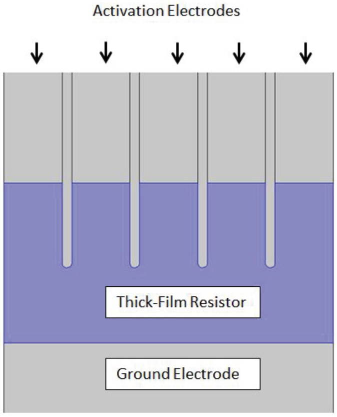
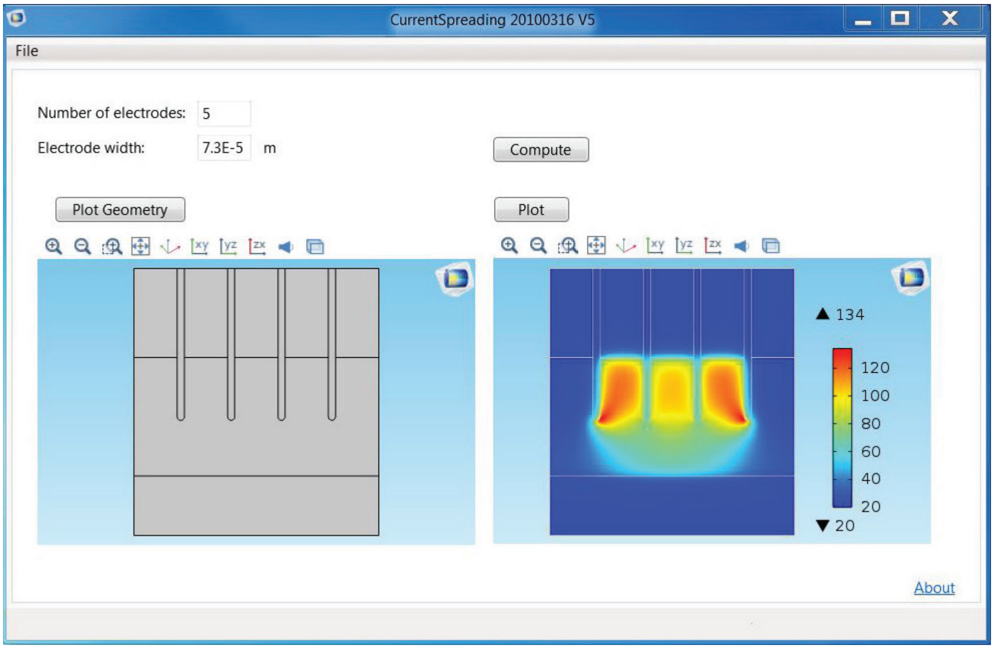


FIGURE 1. Idealized print head schematic.

Voltage may be applied to selected electrodes of the print head to heat corresponding portions of a resistive film, and the purpose of the model is to discover the final temperature distribution in the film.

To implement the model as an app, I used the Application Wizard. The wizard starts with a COMSOL® software model and assembles lists of elements from the model that are suitable for use as inputs and outputs, modeling operations, and graphics.

I selected the number of electrodes and their width as inputs, a command that constructs the geometry and another that executes the model as operations, and plots of the



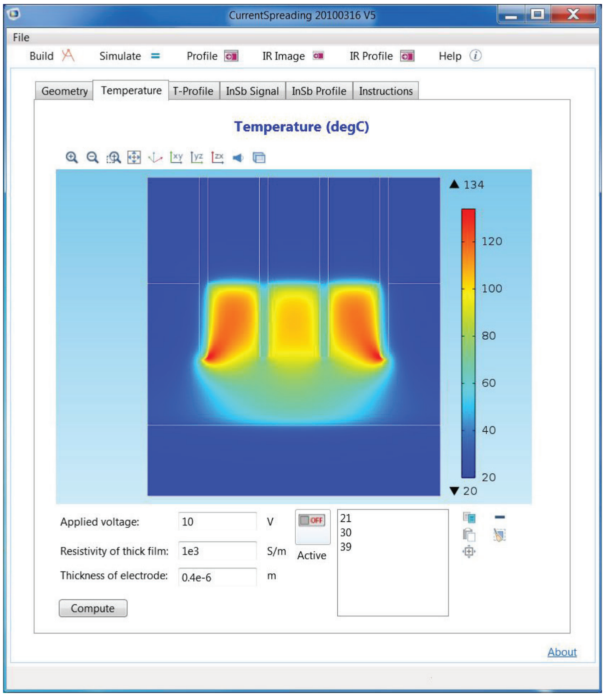
Number of electrodes: 5
Electrode width: 7.3E-5 m

FIGURE 2. A screenshot of the app I built using the Application Wizard.

Abridged piece, originally published in February 2015 on the *Physics Today* website.

object geometry and the surface temperature as graphics.

With these selections, the result was a working app that has input or display boxes for the input/output data, buttons for the actions, and display panels for the graphics. The app components, or widgets, are displayed on a canvas, and can be dragged with the mouse to suitable locations for an attractive interface. A *Test Application* button allows me to launch the new app and test its features from within the COMSOL design environment. Figure 2 shows the app I built from my model using the Application Wizard.



Applied voltage: 10 V

Resistivity of thick film: 1e3 S/m

Thickness of electrode: 0.4e-6 m

21
30
39

FIGURE 3. Apps built using the Application Wizard can be embellished later with forms and other enhancements.

➔FULL POWER AHEAD

The Application Wizard is quick and produces a user-friendly version of a model in just a few minutes, but it does not exercise the full power of the Application Builder. If I want to build something a bit more impressive, an array of tools can help assemble fully featured apps. For that purpose, the Application Builder has two sub-tools: the Form Editor and Method Editor.

The Form Editor is primarily used to generate the page layouts that comprise an app. It offers a wide selection of graphical user interface widgets, and users specify sequences of actions associated with each of them. Examples include buttons, check boxes, combo boxes, radio buttons, text boxes, sliders, and tables. The Form Editor also offers a progress bar, a message log, an equation widget displaying formatted LaTeX equations, and a results table.

The second sub-tool is the Method Editor, which is essentially a Java® programming environment that allows users to combine the COMSOL® interface with Java® programs and libraries in order to add additional functionalities.

The Method Editor also has capabilities related to modeling. As a Java® programming environment, it can, of course, implement Java® code, classes, and libraries from any source. However, it can also interact with the COMSOL model tree and the *COMSOL API* in a number of useful ways. For example, a recording feature allows users to turn on the recorder, carry out operations in the model tree (such as creating a graph, or specifying a mesh), and then turn off the recorder to find the equivalent Java® code added to the method.

➔FURTHER EMBELLISHMENTS

After attending a COMSOL workshop on the Application Builder and reading *Introduction to Application Builder*, I spent a day embellishing the original app. The new app, shown in Figure 3, has a toolbar at the top and has the tasks of geometry definition, model execution, and display of results divided into separate pages. The first of the tabbed windows allows specification of the print head pixel count and dimensions, and then draws the print head (the geometry is also constructed and resized to fit in the graphics window).

The second tab of the app, which is shown in Figure 3, has two functions. The selection box produces a drawing of the geometry in the graphics window. This is a live drawing, and clicking any of the electrode boundaries results in the application or removal of a voltage on that boundary. Clicking the *Compute* button carries out the meshing and solving operations during which a progress bar appears on the lower right to track the progress.

On completion, the model sounds a chime and prepares several report pages with further results: One is a 1D cut through the 2D surface temperature data to show the temperature profile through the centers of the pixels, while two others normalize the temperature data and apply the typical response curve of an indium antimonide detector.

This is the distillation of my few hours with the Application Builder. Creating an easy-to-use interface is not an onerous task, and there is satisfaction, even for a skilled model builder, in seeing the clutter removed before setting about using a model productively.

The major contribution of the new tools added in COMSOL Multiphysics 5.0 is the ability to integrate the development into a single tool, to greatly simplify the interaction with the API, and to provide a standalone server that is separate from the detail-oriented model development tools. This change promises to make the user interface a more natural and commonplace part of the model development process, and to engage a much wider audience in the use and appreciation of multiphysics models. ❖

William T. Vetterling is a research fellow and manager of the Image Science Laboratory at Zink Imaging, as well as a co-author of the Numerical Recipes *series of books and software.*

Oracle and Java are registered trademarks of Oracle and/or its affiliates.