Commercial Off-The-Shelf (COTS) Components in Software Engineering: The Software Package SCILAB

N. Gnanasankaran  
MCA Department, K.L.N. College of Information Technology, Pottapalayam-630 611, India, sankarn.iisc@gmail.com

S. Natarajan  
School of Physics, Madurai Kamaraj University, Madurai-625 021, India, s_natarajan50@yahoo.com

K. Alagarsamy  
Computer Centre, Madurai Kamaraj University, Madurai-625 021, India, alagarsamy.mku@gmail.com

K. Iyakutti  
Department of Physics & Nanotechnology, SRM University SRM Nagar, Kattankulathur-603203, India, iyakutti@gmail.com

Abstract

Commercial Off-The-Shelf (COTS) components are widely used in many software industries and also in scientific computing. This paper first considers the definition of the term COTS and then tries to find how it typically manifests itself in the softwares used for mathematical computations. Some details regarding one of the popular software used for such calculations, viz., SCILAB are furnished. Since this is a freely downloadable program and a non-commercial one, it is better to consider this as an example of Scientific Off-The-Shelf (SOTS) components, a newly coined acronym.

Keywords: COTS, Computations, Mathematics, Programs

2. Softwares for Mathematical Computations

Numerous software packages are available for mathematical calculations in several branches of science. Some of the popular packages are the following: MAGMA, MAPLE, Mathematica, MATLAB, MAXIMA, MLAB, REDUCE, SAGE and SCILAB. In the following sections, some of the features of the software: SCILAB are discussed. This package is considered since it is a free software whereas MATLAB is a paid one; also SCILAB’s computational capabilities are almost equivalent to that of MATLAB.

3. Overview of SCILAB

All the details furnished below are from the “Instruction Manual” of the program manual [4]. Scilab is a mathematical package associated with a rich collection of numerical algorithms covering many aspects of scientific computing problems. From the software point of view, Scilab is a dynamical program package. This generally results in faster development processes, since the user directly accesses a package, with a rich set of features provided by the library. The Scilab package is meant
to be extended so that user-defined data types can be defined with possibly overloaded operations. Scilab users can develop their own modules so that they can solve their particular problems. The Scilab package allows to dynamically compile and link other languages such as FORTRAN and C: this way, external libraries can be used as if they were a part of Scilab built-in features. Scilab also interfaces LabVIEW, a platform and development environment for a visual programming language from National Instruments.

From the license point of view, Scilab is a free software in the sense that the user does not pay for it and Scilab is an open source software, provided under the Cecill license [5]. The software is distributed with source code, so that the user has an access to Scilab's most internal aspects. Most of the time, the user downloads and installs a binary version of Scilab, since the Scilab consortium provides Windows, Linux and Mac OS executable versions. Online help is provided in many local languages. From the scientific point of view, Scilab comes with many features. At the very beginning of Scilab, features were focused on linear algebra. But, rapidly, the number of features extended to cover many areas of scientific computing.

The following is a short list of its capabilities:
Linear algebra, sparse matrices, polynomials and rational functions, interpolation, approximation, linear, quadratic and non-linear optimization, ordinary differential equation solver and differential algebraic equations solver, classic and robust control, linear matrix inequality optimization, differentiable and non-differentiable optimization, signal processing, statistics.

Scilab provides many graphics features, including a set of plotting functions, which allow to create 2D and 3D plots as well as user interfaces. The Xcos environment provides a hybrid dynamic systems modeler and simulator.

4. How to get and install Scilab

Whatever your platform is (i.e. Windows, Linux or Mac), Scilab binaries can be downloaded directly from the Scilab homepage: [http://www.scilab.org](http://www.scilab.org) or from the Download area: [http://www.scilab.org/download](http://www.scilab.org/download). Scilab binaries are provided for both 32 and 64-bit platforms so that they match the target installation machine. Scilab can also be downloaded in source form, so that you can compile Scilab by yourself and produce your own binary. Compiling Scilab and generating a binary is especially interesting when we want to understand or debug an existing feature, or when we want to add a new feature. To compile Scilab, some prerequisites binary files are necessary, which are also provided in the download center. Moreover, a FORTRAN and a C compiler are required.

4.1 Installing Scilab under Windows

Scilab is distributed as a Windows binary and an installer is provided so that the installation is really easy. Several comments may be made about this installation process. On Windows, if your machine is based on an Intel processor, the Intel Math Kernel Library (MKL) [6] enables Scilab to perform faster numerical computations.

4.2 Installing Scilab under Linux

Under Linux, the binary versions are available from Scilab website as .tar.gz files. There is no need for an installation program with Scilab under Linux: simply unzip the file in one target directory. Once done, the binary file is located in `<path>/scilab-5.x.x/bin/scilab`. When this script is executed, the console immediately appears and looks exactly the same as on Windows.

Notice that Scilab is also distributed with the packaging system available with Linux distributions based on Debian (for example, Ubuntu). This installation method is extremely simple and efficient. Nevertheless, it has one little drawback: the version of Scilab packaged for your Linux distribution may not be up-to-date. This is because there is some delay (from several weeks to several months) between the availability of an up-to-date version of Scilab under Linux and its release in Linux distributions. For now, Scilab comes on Linux with a binary linear algebra library which guarantees portability. Under Linux, Scilab does not come with a binary version of ATLAS [7], so that linear algebra is a little slower for that platform, compared to Windows.

4.3 Installing Scilab under Mac OS

Under Mac OS, the binary versions are available from Scilab website as a .dmg file. This binary works for Mac OS versions starting from version 10.5. It uses the Mac OS installer, which provides a classical installation process. Scilab is not available on Power PC systems.

Scilab version 5.2 for Mac OS comes with a Tcl / Tk library which is disabled for technical reasons. As a consequence, there are some small limitations on the use of Scilab on this platform. For example, the
Scilab / Tcl interface (TclSci), the graphic editor and the variable editor are not working. These features will be rewritten in Java in future versions of Scilab and these limitations will disappear.

Still, using Scilab on a Mac OS system is easy, and uses the shortcuts which are familiar to the users of this platform. For example, the console and the editor use the Cmd key (Apple key) which is found on Mac keyboards. Moreover, there is no right-click on this platform. Instead, Scilab is sensitive to the Control-Click keyboard event. For now, Scilab comes on Mac OS with a linear algebra library which is optimized and guarantees portability. Under Mac OS, Scilab does not come with a binary version of ATLAS [7], so that linear algebra is a little slower for that platform.

5. Getting Started

There are several ways of using Scilab and the general methods are the following: 1) using the console in the interactive mode, 2) using the exec function against a file, 3) using batch processing. The details of these methods are not dealt with in this paper.

6. Basic Elements of the Language

Scilab is an interpreted package, which means that it allows to manipulate variables in a very dynamic way. In this section, we present the basic features of the package, that is, we show how to create a real variable, and what elementary mathematical functions can be applied to a real variable. If Scilab provided only these features, it would only be a super desktop calculator. Fortunately, it is a lot more and this is the subject of the remaining sections, where we will show how to manage other types of variables, that is booleans, complex numbers, integers and strings.

It seems strange at first, but it is worth to state it right from the start: in Scilab, everything is a matrix. To be more accurate, we should write: all real, complex, boolean, integer, string and polynomial variables as matrices. Lists and other complex data structures are not matrices (but can contain matrices). This is why we could begin by presenting matrices. Still, we choose to present basic data types first, because Scilab matrices are in fact a special organization of these basic building blocks.

In Scilab, we can manage real and complex numbers. This always leads to some confusion if the context is not clear enough. In the following, when we write real variable, we will refer to a variable whose content is not complex. In most cases, real variables and complex variables behave in a very similar way, although some extra care must be taken when complex data is to be processed.

7. Elementary Mathematical Functions

A large number of elementary mathematical functions are available. Most of these functions take one input argument and return one output argument. These functions are vectorized in the sense that their input and output arguments are matrices. This allows to compute data with higher performance, without any loop.

7.1 Booleans

Boolean variables can store true or false values. Several comparison operators are available in Scilab. These operators return boolean values and take as input arguments all basic data types (i.e. real and complex numbers, integers and strings).

7.2 Complex Numbers

Scilab provides complex numbers, which are stored as pairs of floating point numbers. The pre-defined variable %i represents the mathematical imaginary number i which satisfies $i^2 = -1$. All elementary functions, such as $\sin$ for example, are overloaded for complex numbers. This means that, if their input argument is a complex number, the output is a complex number.

7.3 Function Libraries

A function library is a collection of functions defined in the Scilab language and stored in a set of files. When a set of functions is simple and does not contain any help or any source code in a compiled language like C/C++ or FORTRAN, a library is a very efficient way to proceed. Instead, when we design a Scilab component with unit tests, help pages and demonstration scripts, we develop a module. Developing a module is both easy and efficient, but requires a more advanced knowledge of Scilab. Moreover, modules are based on function libraries, so that understanding the former allows to master the latter. Modules will not be described in this document. Still, in many practical situations, function libraries allow to efficiently manage simple collections of functions.
7.4 Plotting
Producing plots and graphics is a very common task for analysing data and creating reports. Scilab offers many ways to create and customize various types of plots and charts. Scilab can produce many types of 2D and 3D plots. It can create x-y plots with the plot function, contour plots with the contour function, 3D plots with the surf function, histograms with the histplot function and many other types of plots.

8. Computations Involved
Suitable computer programs have been developed for the different types of mathematical calculations arising in the various branches of science. The modern software packages incorporate all the above programs in a unique way and help the scientists to obtain the solutions for their problems. In general, no one knows the details of the programs used in the above packages and they remain black boxes, but help the scientists! In the case of the package: Scilab, it is not completely a black box, as could be seen from the information given already about this package.

9. Conclusion
One could easily visualize the usage of Commercial Off- The- Shelf (COTS) technology in the above example. The modern packages are very versatile in their functioning and the quality is not compromised in utilizing the above software package. Hence, the above could be considered as an example for the positive utilization of COTS components, in designing a software package for the usage of a group of scientists. The program package discussed above is freely downloadable from websites and a non-commercial product. It is suggested that examples of these nature may be categorized as Scientific Off-The-Shelf (SOTS) components, a newly coined acronym.

10. Acknowledgement
One of the authors (NG) thanks the Management and the Principal of the K.L.N. College of Information Technology, for the encouragements.

11. References