An Outline of the Athena Tools Library
Draft for Comments

David Flanagan, Chris Craig

July 30, 1990

Overview

The Athena Tools library will be a collection of useful and usable tools for the development of technical courseware at Project Athena. It is intended to provide, in a well documented, reliable, centrally available, and generally useful form, solutions to the most common hurdles developers of technical software face in our environment.

The library will contain X Toolkit widgets[1] to input and display numeric data on graphical displays, as well as non-widget facilities that simplify data analysis, aid in the creation of simulation programs, and provide structured graphics and hypertext capabilities. The list below gives an overview of the elements of the library. The remaining sections of this document provide a more detailed description of each item. Please note that this document describes a preliminary design of the Athena Tools library. The specifications are intentionally informal, and in some cases vague. Implementation of some modules is underway, but for the most part, the widgets and objects described do not yet exist.

Comments are solicited; send them to the Project Athena Faculty Liasons. Hardcopy to E40-343A, or electronic mail to *f_l@athena.mit.edu.*

AtText Routines to display formatted text and mathematical symbols.

AtLabel A simple widget that uses the AtText routines to display formatted text.

AtSlider A widget for display and editing of a numeric quantity. Similar to the sliding potentiometers on a graphic equalizer.

AtParameter A widget that displays the name, value and units for a numeric parameter and allows the user to click to increment or decrement the value.

AtParameterBox A widget that allows for the convenient creation and layout and use of groups of AtParameter widgets.

AtTable A widget that displays and allows editing of numbers and strings in tables.

AtStripchart A widget that simulates a chart recorder or oscilloscope.

[1] A "widget" in X Toolkit terminology is a user-interface object, a building block of user-interfaces.

AtPlotter A widget that displays 2D x-y and contour plots

AtMeshPlot A widget that displays 3D mesh plots

AtScratchpad A widget that allows the user to evaluate symbolic mathematical expressions and define variables and functions.

AtCli A "command line interface" widget. Applications can easily print out prompts and read user input.

AtPicture A widget to display drawings.

AtHelp A widget for browsing on-line help information.

Typed arithmetic routines A set of common mathematical data types and procedures to do arithmetic on them. Complex numbers, experimental data (numbers with error bars), dates, times, etc., are supported.

Symbolic expression evaluation routines Routines that make use of the typed arithmetic facilities to do symbolic math. Algebraic expressions are parsed and evaluated. Quantities can be assigned to variables.

Symbol table object An object that holds name, value pairs for an application. Useful in simulation programs, and with the AtParameter widget.

Model object The "back end" of the table widget. In combination with that widget it provides the functionality of a powerful spreadsheet program. Used alone it is useful for data analysis and modeling.

Drawing routines "object oriented" drawing routines. Pictures are composed out of drawing primitives or other sub-pictures. Pictures can be translated, rotated and scaled.

Notification handler This module is in charge of telling other objects when quantities they are interested in change.

Dialog management routines Routines to automatically create dialog boxes and return the user's responses to the program.

Scaling routines Routines to convert between user and pixel coordinate space. Used by several of the widgets

Scope of the Library

The Athena Tools library is a collection of functions callable from C. It does not contain CASE tools such as a user interface builder. While some components of the Athena Tools hint at a far reaching "program development environment", this library is not such an environment itself.

AtText

The AtText data structure and associated routines allow for the display of formatted multifont text as well as greek letters and mathematical symbols. Superscripts and subscripts are supported as are square-roots, fractions, integrals, summations etc. Multi-line text is supported, and there are routines to do automatic word wrapping of the text. Text may be displayed rotated 90 degrees counterclockwise. It is possible to specify parts of the formatted text as "hot spots", and determine when a user has clicked the mouse over these areas of the text. This provides a basic hyper-text capability. All text is displayed in fonts belonging to a single specified "font family". There are routines for creating and manipulating font families. Text is formatted for display with embedded commands similar to those used by the Scribe text formatter. The AtText routines operate at the Xlib level. Most of the widgets in the library use these text display routines.

AtLabel

This widget is very similar to the Motif label widget, except that it uses the AtText facilities to allow the display of formatted text.

AtTable

This widget displays and allows the editing of tables of numbers and strings. It allows the user to cut and paste rows, columns, or regions of the table, and there are PostScript output routines to produce hardcopy of the table. Alone, the AtSpreadsheet widget has none of the traditional mathematical spreadsheet capabilities. These are the domain of the AtModel object, which is designed to work well with the AtTable widget as its front end.

AtPlotter

The AtPlotter widget displays 2-dimensional x-y axes and will plot data points, optionally connected, and optionally with error bars. It will also compute and display contour plots of 3-dimensional data. The axes can be linear or logarithmic. The plot can be given a title, the axes labeled, and the plot annotated with text. There is an optional legend. Individual plots can be selected by clicking on their name in the legend. A selected plot can be pasted into other windows or applications. The widget can generate presentation quality postscript output of the axes and the plots. Each plot is treated as a widget itself, and new plot types can be added by subclassing an existing type.

AtStripchart

The AtStripchart widget simulates a chart recorder or an oscilloscope. It is different from a 2D plotting widget in that it displays data point-by-point as it arrives, rather than an entire plot at a time. Any number of "traces" can be displayed, and a scrollbar can be attached to examine data that has scrolled sideways off the screen. The user can drag the baseline of individual traces up or down as convenient, and can use the mouse to measure relative distance on the screen — this allows measurement of the period, frequency, and amplitude of a trace. Individual traces can be selected and used with cut-and-paste. The widget can be configured to poll for data at a fixed time interval, or to simply display data as the application generates it.

AtMeshPlot

The AtMeshPlot widget displays 3-dimensional mesh plots on a set of x,y,z axes. Hidden lines will be removed, and the widget will allow the axes to be rotated, and the angle of the observer above the plane to be changed. The user will be able to select points in 3-space and cross-sections of the data, probably through some sort of mouse interaction with the axes. The widget will support PostScript output.

AtSlider

The AtSlider widget displays what amounts to an x or y axis from a 2-dimensional plot. It displays a numeric value with a special mark on the axis, and allows the user to change that value by dragging the mark with the mouse. It can be used in the same capacity as a sliding potentiometer is used on physical equipment. The Motif widget set provides a similar widget, XmScale, but it only allows integer values.

AtParameter

The AtParameter widget displays the name, the value, and the units for a single "parameter". It also provides two arrow buttons which the user can click on to increment or decrement the displayed value. The user may also use the keyboard to directly set the value of the parameter. This widget is useful in simulation applications, for example, which require the user to input several numeric parameters to the simulation engine. They are compact and require little screen space.

AtParameterBox

The AtParameterBox is designed to provide a convenient way to create and display multiple AtParameter widgets. The AtParameterBox will automatically create the individual AtParameter widgets, if desired, and the programmer can look up parameter values by name. The ParameterBox will be optionally scrolled, and groups of related AtParameter widgets can be graphically set off from one another.

AtScratchpad

The AtScratchpad widget provides a text area in which the user can define variables and functions and evaluate symbolic expressions. Constants and functions can be pre-defined by the application writer. This widget uses the expression evaluation routines described below. Properly setup, the scratchpad could allow the user to set the value of numeric parameters displayed in AtSlider or AtParameter widgets.

AtPicture

The AtPicture widget is a container for the drawing objects described below. It manages a display list of objects which it will refresh as needed. It notifies the application through callbacks when the user selects a drawn item, or clicks, double clicks or drags with the mouse. It will rubber-band a box as the user drags the mouse. This allows the program to respond to user interaction with the drawing objects that make up the picture.

AtHelp

The AtHelp widget will display on-line help for an application. It will support multi-font, hierarchically organized text with hypertext "hot spots" for following cross-references.

AtCli

The AtCli widget is a "command line interface" for applications that want a terminal based interface. It is similar to working with an xterm in an application — the application can easily "print" out characters, and can get a callback when the user enters a line and hits the return key. It provides cut-and-paste of text, scrolling, and command history using the up and down arrow keys.

Typed Arithmetic Routines

This part of the library will define numeric data types and operators and functions to operate on them. Complex numbers and experimental data (data points with errors) will be defined with a complete set of arithmetic and trigonometric functions. Random variables, dates, times, and other types may also be defined.

Expression Evaluation Routines

These routines will provide a symbolic expression parsing and evaluation interface. All the data types described above will be supported. Expressions will be evaluated in the context of a given symbol table. There will be a method of assigning a value to a variable in a symbol table, and of defining single and multi-variate functions. A formatted text string suitable for display with the AtText routines can be extracted from a parsed expression.

Symbol Table Object

The symbol table object is used with the expression evaluation routines and the scratchpad widget to store the defined values of variables. Symbol tables can be created as children of other symbol tables, and variables defined in a child override those in the parent. When creating a symbol table the programmer can predefine values for any desired variables.

Table Object

The table object is designed as a standalone data analysis or modeling object, or to serve as the back-end to the AtSpreadsheet widget. It allows manipulation of data organized into rows and columns. Rows, columns, or cells of the data can have symbolic expressions associated with them — the values for that row, column, or cell is obtained by evaluating the expression. Rows, columns, and sub-ranges can easily be extracted from the table and used with the AtPlotter widget, for example.

Drawing Objects

These are a set of objects that draw lines, arcs, rectangles, ellipses, splines, etc. They know how to scale, rotate, and translate themselves, and can be composed to form compound drawing objects. The ability to create composite drawing objects makes it easy to define primitives for any sort of domain-specific drawing task. They are useful with the AtPicture widget. Each of the objects can draw itself in an X window and output a postscript representation of itself.

Notification Handler

A module used with the Symbol Table object. The application or other widgets or objects can register interest in a particular symbol and will be notified via callback when the value of that symbol changes.

Dialog Management Routines

These routines, yet to be designed, will allow the programmer a relatively simple way to create dialog boxes and will return the user's responses in a way that is convenient for the program.

Scaling Routines

Routines used by several of the objects in the library. They convert between pixel space and user space and vice versa. They support linear and logarithmic transforms.