PRODUCT OUTLINES for MATHEMATICA FRONT-END PROJECT

William Bricken
September 1988

CONTEXT

Mathematica was released for alpha testing early in 1988. Very few people saw the product before its public release on 6/23/88; most of these folks were academics. We were able to evaluate and familiarize with Mathematica in 4/88. We have therefore created a market opportunity, being virtually the first third-party developers of Mathematica front-end enhancements. This opportunity with close quickly, due to the overwhelming public enthusiasm for Mathematica.

PRODUCT OUTLINES

We propose three potential front-end projects:

1. Engineering Spreadsheet (ES), an adaptation of traditional spreadsheet display technology to a Mathematica engine,

2. SketchMatica (SM), the union of an interactive graphics editor and Mathematica, which would provide both a graphic front-end for display of Mathematica computations, and an interactive visual driver for the Mathematica engine, and

3. Visual Language (VL), an application of boundary mathematics to the display and specification of Mathematica computations.

The three potential projects are hierarchically interrelated. Both the Engineering Spreadsheet and SketchMatica are subsets of the Visual Language project. ES provides display of array and matrix data-types, while SM provides the rudimentary visual specification capabilities for Mathematica programs. SM also provides the display substrate for the ES.

The ES project has the short range goal of providing a fairly rapid enhancement to Mathematica. It requires a configurable display of tables, and could be based on either character graphics (like traditional spreadsheets) or pixel graphics.

The SM provides a window-based interactive graphics capability for Mathematica. Uses include display of spreadsheet tables, display substrate for high level graphics languages (such as Logo), symbolic constraints on graphic objects, and visual display of computation. The SM can be considered as a toolkit for graphic display, and could include easy specification and generation of buttons, sliders, etc. for graphic display controls. It also provides graphic (WYSIWYG) specification capability.

The VL project has the long-term goal of providing an interactive specification and development language based on visual representations. Restrictions on the style and expressability of the language are necessary to keep the project tractable. The VL is a long-range research project that supports modular releases of products, such as the ES and the SM.

The ES and the SM are solely a Mathematica enhancements, but the VL is both a front-end for Mathematica, and a stand-alone product that will be attachable to various target languages, such as Mathematica, Pure LISP, Pure PROLOG, and other mathematically structured modern languages. The integration of the VL and Mathematica is close in that both use the same mathematical techniques to achieve computation (equational substitution, function composition, operator calculus, single binding semantics, ...)

The VL has a yet wider scope, since the formalism which underpins the visual representation has many visual isomorphs (there are several mathematically equivalent visual languages), is realizable in a parallel architecture (visual substructures can be distributed across processors), is a common representation over target languages (cross-translation and transparent parallelization are possible), supports imaginary and contradictory specifications without degradation, and can make several small angels dance on the head of a pin.

CONTEXT for FRONT-END DESIGN AND DEVELOPMENT

Most of the functions that an engineer requires are built-in function calls in Mathematica. What is lacking is:

1. a friendly display of array and matrix data structures (the native storage mechanisms in Mathematica),

2. a convenient organizational model to help users arrange and structure series of calculations (Currently users must scroll over input and output lines, and must have a coherent plan of attack for composition of computations. An alternative is to write Mathematica programs. However, the programming language is quirky, and the environment is impoverished.),

3. a visual rather than line-based display (Although output is two-dimensional mathematical notation, it is space-consuming and difficult to read.), including image editing, and

4. interactive graphics.

EVALUATION of POTENTIAL FRONT-END ENHANCEMENTS

Matching the capabilities of our front-end to those of the kernel will be difficult. Since Mathematica is so sophisticated computationally, it will accentuate weaknesses in the interface. Therefore the front-end design should focus on interface rather than computational functionality.

I prefer to focus on the SM rather than the ES as a product for these reasons:

1. The enhanced functionality offered by the ES is dubious. We do not know how an engineer would use a spreadsheet to enhance the powerful functions available in Mathematica.

2. The ES project does not integrate well with other lab projects. The SM, on the other hand, is closely related to our VL projects, to the Turtle Graphics and fractal generator experiments, and to current work on graphics toolkits.

3. Using a graphics front-end for Mathematica would enhance existing as well as new markets.

4. The SM provides prototype information for future CAD enhancements and VL development.

SPECIFICATION OUTLINE for the ENGINEERING SPREADSHEET

Conformity to standard spreadsheet look-and-feel, and expected functionality.
Visual display of array and matrix data structures.
Ability to localize computation in any cell or row.
Ability to display steps in a computational sequence.
Automatic re-computation of cells in rows that are functionally connected.
Graphs as cells.
Configurable cell sizes.
Documentation in Mathematica Notebook form.
Spreadsheet configuration and specification language.

The following specifications are provided directly by Mathematica. Our front-end enhancement is to make display and access to these capabilities easier.

Symbolic data in cells.
Arbitrary precision computation.
Array and matrix computation.
Optional numerical or symbolic computation.
Power series as arrays.
Logical constraints on cells.

SPECIFICATION OUTLINE for SKETCHMATICA

Interactive graphic display of Mathematica output.
Tabular display of stored data, including ES display capabilities.
Mouse entry of points for Mathematica analysis.
Menu entry of graphic object attributes and constraints.
Mathematica programmability of display.
Graphic editing and specifications.

SPECIFICATION OUTLINE for the VISUAL LANGUAGE

The project is composed of modular Phases:

Phase I: Display of algorithms, animation of running algorithms.
Phase II: Interactive construction and debugging of visual programs.
Phase III: Visual tools, transparent operating environment.

PHASE I SPECS

Stand-alone or as Mathematica front-end
Displays Prolog, LISP, and Mathematica program structure, control structure, and data structure.
Animated display of program execution and flow of control.
Animated display of database structure and search.
Animated display of function invocation, expansion, and application.
Developed for a memory-rich environment (MacII, SUN).

PARTIAL PHASE II SPECS

Mouse-driven construction of visual programs.
Interactive editing of visual structures.
Interactive composition of program fragments.
Evaluation of partial functions.
Stepping and trace of control flow.

DEVELOPMENT ENVIRONMENT

The target machine for the Mathematica front-end will support OS2. However, development will need to proceed in SUN and Mac environments until the Mathematica OS2 port is near completion.

The ES would probably be tailored for IBM machines, but the SM could be developed for SUN and Mac machines also.