Project Description

techexplorer and MathDL: Robust Support for Dynamic Web-based Mathematics

Statement of Need

The Mathematical Sciences Digital Library, or MathDL, is an NSDL-affiliated collection of learning materials for mathematics. As with the NSDL, MathDL is much more than a collection; it is a community that seeks to understand how to create, deliver, and use digital resources for effective learning. A major focus of MathDL is *interactivity* because of its tremendous potential for improving mathematics education. Many students have difficulty learning mathematics: they must compute and visualize the results simultaneously, while at the same time communicate in the difficult, foreign language of mathematics. Interactivity can support computation, visualization, and language in any combination. By changing the mode of interaction, students and teachers can isolate the relationships between computation and language, between language and visualization, or between language and computation; thus, they can more readily make the connection among the many elements of mathematics.

There are several ways to deliver interactive computations over the web: scripting languages support most arithmetic operations; Java applets can deliver computations that are constrained to specific operations; and computer algebra systems are available as web services for open-ended computation. Java provides robust interactive graphing capabilities in one and two dimensions, and several efforts are underway to develop OpenGL-enabled Java implementations that will provide interactive graphing in three dimensions. Adequate web support for interactive mathematical *notation*, however, is severely lacking.

Mathematics of any sort is difficult to render electronically, with its reliance on special characters, two-dimensional layouts such as fractions and matrices, and pervasive use of domain-specific and author-defined notation. Although mathematics itself seeks to be axiomatic, the language of mathematics is most irregular, with each subdiscipline of mathematics often having its own alphabet, vocabulary, grammar, and syntax. Yet this is how we communicate mathematics, and part of what makes learning mathematics so difficult.

MathML: A Standard Protocol for Mathematics Supporting mathematics notation interactively is especially difficult because the computer must understand both syntax and semantics in order to compute. To address the disparity between the two, members of the World Wide Web Consortium (W3C) developed the Mathematical Markup Language, or MathML, as a standard XML markup that gives applications a common language in which to communicate semantics, while also providing a method for mapping from semantic expressions to their various visual representations. MathML takes two forms:

- *Presentation MathML* describes the visual layout of mathematical expressions. Because the same mathematical notation can mean very different things in different contexts, Presentation MathML can be misinterpreted if taken out of context. The Greek letter Γ, for instance, could represent an arbitrary variable in physics, the "Euler gamma function" in number theory, or the "gamma distribution" in statistics—none of which are related.
- *Content MathML* describes the semantic meaning behind mathematical expressions that is necessary for computation. Content MathML is semantically unambiguous, but it says nothing about how an expression should appear visually. Consequently, a transformation to Presentation MathML or another typographic format is required to render Content MathML.

In order to exchange mathematical data seamlessly, computer applications that support MathML need to support both forms: Presentation MathML for human interactions, and Content MathML for computer interactions.

MathML offers tremendous potential for resolving the disparity between syntax and semantics, and we are now seeing broad support for MathML in mathematics applications. Several computer algebra systems now support the import and export of MathML; and two of these—Maple and Mathematica—can be accessed as web services using MathML as a common data format. Most combinations of web browsers and operating systems now render both Presentation and Content MathML with the aid of browser plug-ins and/or a MathML stylesheet developed by the W3C [1]. Despite these developments, delivering MathML over the web is still difficult and unreliable if one relies on this multitude of delivery mechanisms, each with its own methods for embedding and rendering MathML. Further, support for rendering does not translate into support for interactivity: while the W3C's MathML stylesheet is useful for minimizing browser and plug-in dependencies, its utility is limited to *static* MathML content. The techexplorer plug-in remedies all of these shortcomings; it is a single application that supports both static and dynamic rendering, regardless of browser or operating system.

Extending Support for MathML Although rendering static mathematics notation in web browsers is difficult in itself, the term *interactive mathematics* should also extend to notation. What exactly do we mean by *interactive mathematics notation*? In general, we mean two things: scripted expressions and equation editing.

Scripting languages, such as JavaScript, can be used to rewrite portions of an equation in response to some action on the part of the user, such as clicking a mouse or dragging a slider. For instance, this action may update a graph, allowing a student to visualize the relationship between the expression and the graph. Ideally, both graph and expression are updated in real time with the action, and the expression is rendered with the same mathematical notation used, for example, in an accompanying textbook (see Figure 1).

MathML is by no means a necessary part of scripted equations; however, XML's Document Object Model (DOM) provides straightforward methods for identifying

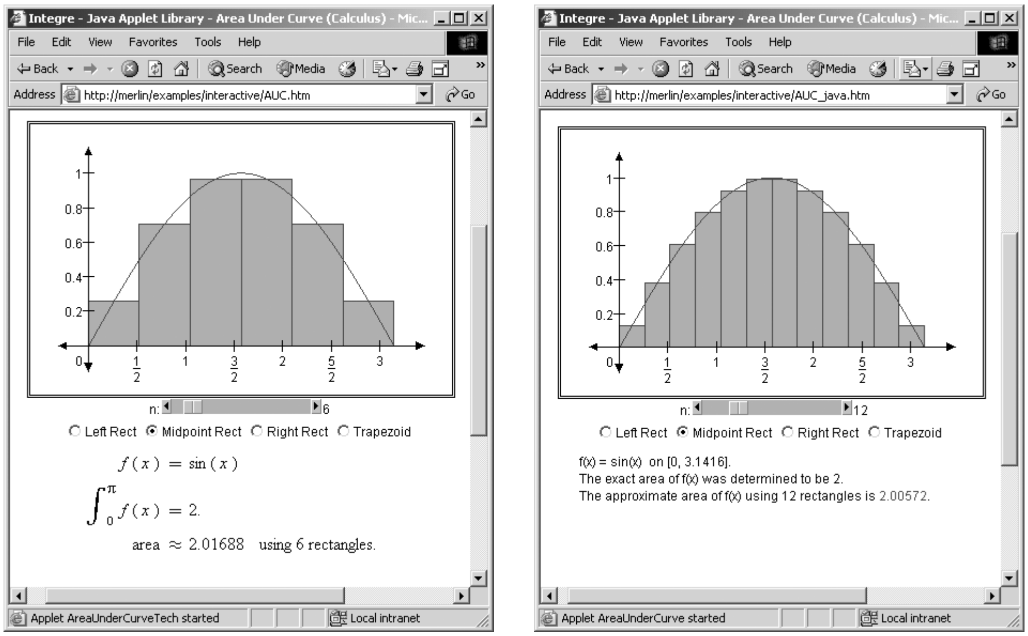

Figure 1. A Java applet demonstrating the calculus concept of approximating the area under a curve. Dragging the slider changes—in real time—both the number of rectangles used to approximate the area and the computed approximation. By using techexplorer it is not necessary to forsake traditional mathematics notation in interactive explorations. *Left:* Mathematics rendered by techexplorer. *Right:* Textual representation of the same mathematics.

and rewriting structural elements or “nodes” within XML data—including MathML equations—without the controlling scripts being embedded *inside* the data. In this way, scripting controls can be separated from the data, and multiple scripting actions can act on the same data. As the web moves increasingly away from HTML and toward extensible XML, the DOM will become an integral part of interactivity. techexplorer uses the DOM internally whenever mathematical expressions need to be modified. Furthermore, techexplorer’s DOM functionality is exposed as an application programming interface (API) so that additional interactive functions can be implemented using the DOM interface.

Equation editing provides a free-form mode of communicating with computers. In some sense, textual input using computer-algebra or calculator syntax can be considered free-form input. However, MathML equation editors provide significant benefits over textual input, particularly in a learning environment:

- Students need not learn a specialized syntax in addition to the traditional notation.
- Equation editors support two-dimensional mathematical notation (e.g., fractions and matrices), which is much easier to read than its one-dimensional equivalent.
- Equation editors can provide constrained editing capabilities merely by reconfiguring palettes or keyboard commands.
- MathML equation editors can communicate with an increasing variety of computational engines.

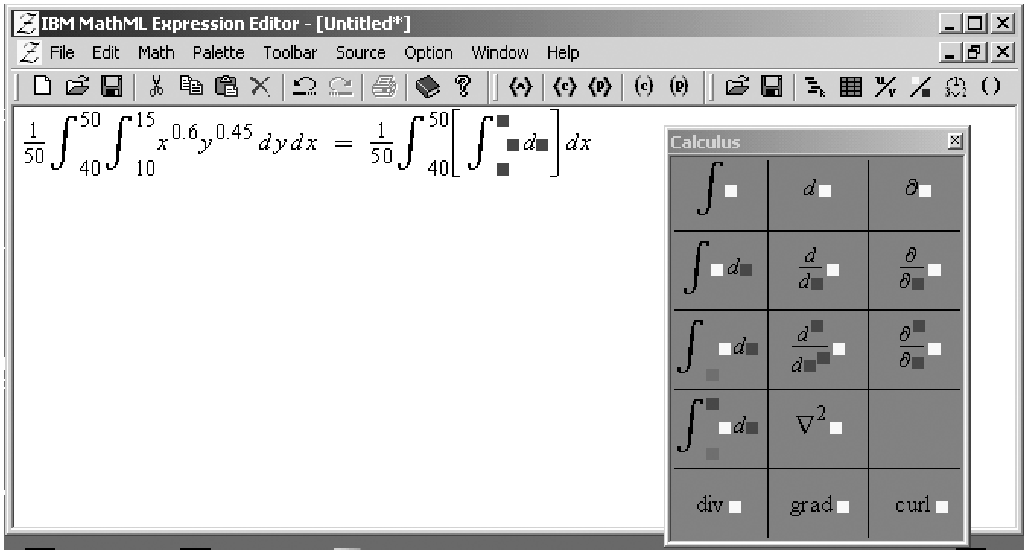

Figure 2. techexplorer's MathML equation editor shown with an unfinished expression and a sample calculus palette.

Only one MathML equation editor is currently available and supported: the WebEQ java applet from Design Science [2]. WebEQ is designed around Presentation MathML, although it also attempts to provide Content MathML with moderate success. Unfortunately, several aspects of WebEQ's design limit its utility: the editing palettes are entirely predefined and cannot be extended to support arbitrary or unique mathematical structures; the editing controls do not strictly enforce the structural rules needed to produce reliable Content MathML; and the applet is large and requires substantial download time.

The techexplorer equation editor was developed toward the end of IBM's involvement and was never released as part of techexplorer (see Figure 2). Nevertheless, this equation editor offers several significant advantages over WebEQ:

- simultaneous creation of Presentation and Content MathML;
- support for user-defined mathematics notation, in the form of customizable mappings between Presentation and Content markup;
- customizable "live" palettes that inherit the default mappings between Presentation and Content;
- user-defined templates that enforce semantically complete (Content) expressions;
- a navigation and selection-control layer that supports multiple editing interfaces (e.g., XML tree navigation versus flat, two-dimensional navigation).

The techexplorer equation editor was implemented (using the DOM interface) as an ActiveX extension to techexplorer, so at present it works only on Windows systems. Part of the development outlined for this project involves moving the equation editor into the plug-in so that it can be used on *all* operating systems. Several competing MathML equation editors are currently in development; however, either they do not support both Presentation and Content MathML [3] or they do not work on all browsers [4].

Summary Most attempts to deliver web-based MathML are inadequate. Using one application for static rendering and another application for interactivity, with different combinations for each browser and operating system, only leads to confusion. Given that the variation in browsers will continue, we have two options: develop a single utility that provides a consistent interface across browsers for both static and dynamic mathematics; or develop enterprise-level server applications for MathML delivery that eliminate browser dependencies entirely. The latter solution, while attractive, is beyond the means of many educational institutions. Most enterprise platforms—most notably Sun's J2EE, Microsoft's .NET, BEA's WebLogic, and IBM's WebSphere—are extremely expensive to purchase, configure, and administer.

The techexplorer browser plug-in provides an appropriate alternative in our quest to support both static and interactive mathematics. It works on all major browsers and operating systems, it supports both static and dynamic mathematics, and it has an extremely robust MathML equation editor. As an added bonus, techexplorer's rendering engine, based on TEX, supports much more complex equation layouts than any other web-based rendering engine (see Figure 3). techexplorer also supports moderately extensive TEX markup, so that mathematics written in TEX does not need to be converted before being delivered by a browser.

Background on techexplorer The techexplorer project began at IBM Research in the mid-1990s as an attempt to render mathematics within web browsers, which at the time were in their own early stages of development. At IBM the early developers were part of the Mathematical Sciences group, and were strongly motivated by their

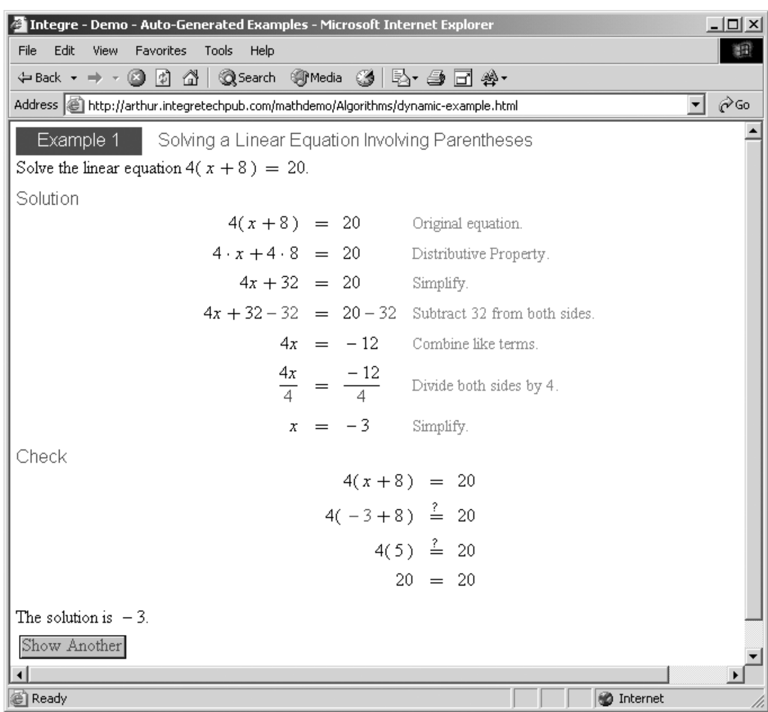

Figure 3. An example of a complex, scripted mathematical layout rendered using techexplorer.

interests in computer algebra and in working with computer algebra systems through web or web-like interfaces. Throughout its development, techexplorer has used XML-compliant methods to implement features whenever possible. Members of the techexplorer development team were co-founders of the W3C's MathML working group, and have served on numerous other standards groups as well. In 1999 the techexplorer project was moved to the domain of IBM's Extensible Technologies group, and for several years techexplorer was often the first application worldwide to implement a new XML standard or technology.

Unfortunately, IBM Research is primarily a research and development organization, and, unlike its software counterpart, lacks any focus on commercialization and the needs of end users. Widespread adoption of techexplorer was impeded by this lack of end-user focus, a situation worsened by IBM's onerous licensing and pricing models. Having been the model for many XML and web technologies, however, techexplorer is a very powerful tool for delivering web-based mathematics. This project seeks to give techexplorer the polish it needs to serve end users appropriately, and to make this technology widely accessible.

Target Audience

The immediate goal of this development project is to support the Mathematical Sciences Digital Library in its mission to disseminate high-quality electronic resources to mathematicians and educators. We are especially interested in giving MathDL's primary audience—instructors, instructional designers, programmers, and electronic textbook authors—the software needed to deliver web-based interactive mathematics. MathDL's content and activities are designed to serve this group, with articles that address the role of interactive mathematics in education, an applet library of high-quality learning objects, and a "developer's area" for discussing best practices in the development of interactive mathematics.

This project also will give MathDL a user-friendly interface to potential web services, such as computation and annotation of MathDL content. Subscriptions to these web services is one method for generating revenue needed to sustain MathDL. In addition, web services provide a means of extending MathDL's target audience. A vast collection of resources can be cost-shared, via inexpensive subscriptions, by institutions that otherwise could not afford them.

Although MathDL specifically targets undergraduate mathematics, we expect that techexplorer's utility will extend to K-12 and graduate-level mathematics, as well as to numerous other disciplines that use mathematics. Ideally, of course, we want techexplorer to benefit anyone who needs web-based mathematics, whether they be in education, industry, or government. These audiences share a need for web-based mathematics, but their computer skills, content interests, and preferred modes of interacting with mathematics vary widely. The design of this project must take into account that diversity.

First, installation of the techexplorer plug-in must be simple and trouble-free. Installation is a user's first experience with software and establishes the user's expectations of software functionality and performance. We need to assume a minimum level of computer literacy; hence installation, configuration, version control, and upgrades must be automated as much as possible.

Second, we need to accommodate users who access the web via low-speed dial-up modems, so minimizing the plug-in download size is important. We will address this concern in three ways: by providing the authoring and programming documentation separately from the plug-in installation; by placing high priority on code optimization; and, where possible, by modularizing plug-in functionality so that upgrades or specialized plug-in extensions can be downloaded piecemeal.

Third, techexplorer needs to support a broad range of mathematical notation, customization of the equation-editing environment, and easy access to metadata. Fortunately, the design of the techexplorer rendering engine and equation editor as XML applications provides for tremendous flexibility and extensibility of these functions, and distinguishes techexplorer's exceptional support for MathML from all other applications. This flexibility will extend techexplorer's utility beyond undergraduate mathematics—to K-12 and graduate-level mathematics as well as to other disciplines that require mathematics. These features are described in more detail later in this proposal.

Fourth, authors need easy paths for moving their mathematics prose into a techexplorer-compatible format. The majority of people who write mathematics use either TEX or Microsoft Word with MathType; a handful use computer algebra systems. MathType supports export to HTML with MathML, which techexplorer can support. Computer algebra systems that support MathML also provide HTML/MathML export with improving results. TEX users have the easiest path of all because techexplorer can render most TEX markup directly without any conversion. (techexplorer does not support TEX's complete macro and programming language.) One significant advantage of using TEX is that authors can fine-tune their documents in native form to achieve the desired look in the browser. Most auto-generated code—including the HTML/MathML exported by Word with MathType and computer algebra systems—requires significant tweaking to achieve satisfactory results, yet the complexity of auto-generated code makes editing that code difficult even for the most expert HTML programmer. We will continue to improve techexplorer's support for TEX markup.

Finally, we know from our experience in commercial software and publishing that content developers will want access to techexplorer features on a variety of levels. Most content developers—in this case, mathematics instructors—are not highly skilled programmers; user-friendly interfaces will provide an alternative means of accessing many of techexplorer's configuration and customization features. Implementing such interfaces will require substantially more time in terms of design and user testing than in actual programming by the project developers. A substantial number of content authors will have competent but not expert programming skills, and they will be comfortable accessing more advanced features using a JavaScript interface or by writing custom configuration files. A few content authors will be expert programmers; they will want unfettered access to techexplorer's most robust features, such as the ability to write "add-in" modules that extend techexplorer's functionality. Extensive documentation will be required to describe the many ways in which developers can create content for techexplorer.

Project Goals

The primary goal of this project is to provide MathDL's audience—especially mathematics instructors—with a single application, namely techexplorer, that supports robust web delivery of mathematics, whether static or interactive. For us to accomplish this, techexplorer must do the following:

- Provide the same features and interfaces on all mainstream browsers (Internet Explorer, Netscape, and Mozilla) and operating systems (Windows, Macintosh, and Linux variants).
- Support rendering and complex layouts of mathematics that is formatted in Content MathML, Presentation MathML, or $\TeX$.
- Provide user-friendly, real-time scripting APIs for mathematics formatted in any of the above-mentioned formats (e.g., Java and JavaScript).
- Expose broader APIs that support computer algebra connectivity, highlighting and annotation, and other web services.
- Provide a highly customizable MathML equation editor that produces both Content and Presentation MathML and allows users to define their own palettes, templates, and mappings between Content and Presentation.
- Support the embedding of valid XML metadata (e.g., attribute tags in MathML) and provide easy user and programming access to the metadata.
- Allow users to copy MathML content from web pages into desktop applications (e.g., computer algebra systems).

In one form or another, all of this functionality exists currently in techexplorer, but has not been implemented for all versions. Also, because techexplorer was primarily a research project, its code base was extended haphazardly and was never optimized. This development project will result in a consistent and optimized plug-in for delivering robust interactive mathematics. techexplorer's ability to support a wide range of mathematics notation will extend its benefits to the larger NSDL community.

A secondary goal of this project is to provide MathDL with an interface with which to deliver mathematics-related web services. Two services are currently being considered: access to computer algebra systems and server-hosted highlighting and annotation of MathDL documents. The techexplorer equation editor can already serve as a wysiwig interface to computer algebra, and the development plan outlined here will implement highlighting and annotation capabilities to techexplorer documents.

Finally, this project seeks to elevate the discussions currently underway regarding "best practices" and the teaching efficacy of web-based mathematics. By collaborating with MathDL and participating in various MathDL forums, we hope to understand better how technology gets used in the classroom, and what improvements are necessary to make that technology better. This understanding combined with our experiences with techexplorer development will help the MathDL community establish a baseline from which best practices in web-based mathematics education can be measured.

Project Design

This project focuses on the continued development of a browser plug-in that provides extensive support for web-based mathematics. As a stand-alone development effort it has the potential to make a substantial impact on mathematics education. The NSDL, however, seeks to go well beyond making an impact by changing the paradigm of science and mathematics pedagogy [5, 6].

To augment the impact of techexplorer development, this project includes a collaboration with MathDL, a thriving community of mathematics educators who are truly out to change the world in their own way. The development undertaken here will be informed by a three-member advisory group whose members are prominent activists within the MathDL community. The guidance of this advisory group combined with the developers' active participation in MathDL forums on web-based mathematics education will make techexplorer a catalyst in the NSDL's efforts to create a new paradigm.

The Advisory Group The purpose of the advisory group is to make sure the development performed under this project addresses in a meaningful way the needs of MathDL users, particularly by enabling the robust delivery of interactive mathematics notation. To establish relevance to the MathDL, the advisory group will do the following:

- Inform the development team with experiences in the classroom and existing MathDL projects, such as the *Journal of Online Mathematics and Its Applications* (JOMA), the Connected Curriculum Project, and the interactive MathDL Books Online.
- Assist software architecture decisions by anticipating future software requirements.
- Offer feedback on usability and reliability of developed software.
- Provide or suggest examples for and feedback on user documentation, prototype projects, and workshop content.
- Contribute as necessary to papers or presentations related to the development project (not necessarily as co-authors, but at least as reviewers).
- Suggest other ways to disseminate the technology and experience.
- At the end of the project, summarize their experiences in a way that may assist the organization of future advisory groups, NSDL technical councils, or other collaborations between digital libraries and software development projects.

The time commitment of advisory group members over the course of the project is expected to be between five and ten days per person.

To launch the collaboration, the principal investigators will meet with the advisory group at the 2003 International Conference on Technology in Collegiate Mathematics (ICTCM). The agenda for this meeting will be to discuss in concrete terms the relevant issues facing MathDL that can be addressed by techexplorer; to establish benchmarks for evaluating techexplorer development per end-user requirements; and to establish a strong working relationship between the development team and the advisory group.

The initial planning meeting will be followed by quarterly status reports to the advisory group. Each quarterly report will serve as a preliminary agenda for a follow-up conference call or face-to-face meeting, the latter in cases where the timing coincides with a conference that is regularly attended by most of the participants. These discussions will provide opportunities to address formally any concerns or questions encountered during development.

To supplement the formal meetings, the developers and advisory group will engage in informal and ongoing discussions about real-world implementation issues. Such interactions may extend to advisory group members seeking limited technical assistance from the developers, or the developers asking for input about whether there is cause for concern about a particular technical or usability issue.

The final project report will include a summary statement that describes and evaluates the relationship between the developers and the advisory group. We hope that this report will inform future technical councils or advisory groups that are created to coordinate the development of technology with the needs of NSDL.

The Development Plan A substantial amount of development time involves redesigning the techexplorer code base to handle more efficiently many of the XML standards and user requirements that were either nonexistent or not evident when techexplorer was originally designed. By maintaining a dedication to XML standards and modularization, this *rearchitecting* effort will make techexplorer much more extensible and easier to maintain as the web evolves.

Following is a task-by-task outline of the development plan. Times are expressed in "programmer months," one programmer month being roughly equivalent to 20 eight-hour days.

- Architecture and Requirements documents (1 month)
- Basic MathML rendering component (3 months)
  - Cross-platform, cross-browser font support (1 month)
  - Core MathML presentation module, node types (1 month)
  - Incremental composition/drawing/rendering support (1 month)
- Extended MathML rendering component (3 months)
  - MathML content extension module, node types (1 month)
  - MathML table extension module, node types (1 month)
  - Version control, auto-detect, auto-download, installation (1 month)
- Basic LaTeX rendering component (3 months)
  - Modular TeX, LaTeX, MathML parsing support (1 month)
  - Core LaTeX presentation module, node types (1 month)
  - Incremental composition/drawing/rendering support (1 month)
- Extended LaTeX rendering component (6 months)
  - Extension modules, node types scoping/design/coding (6×1 month)
- Porting and testing (8 months)
  - Cross-platform, cross-browser printing support (1 month)
  - Cross-platform rendering portability (3 months)

  - Cross-browser rendering portability (1 month)
  - Cross-browser, cross-platform testing (3 months)
- Advanced rendering component (6 months)
  - Standard and custom programming interfaces (1 month)
  - Extension module programming interfaces (1 month)
  - Highlighting and annotation extension module (3 months)
  - CAS connectivity extension modules (1 month)
- Basic MathML editing component (3 months)
  - Core presentation, content node types (1 month)
  - Layouts, templates, keyboard assignments, menus, palettes (1 month)
  - Dialog access to attributes (1 month)
- Extended MathML editing component (3 months)
  - Split-screen presentation/content views (1 month)
  - User friendly customization interfaces (1 month)
  - Customization programming interfaces (1 month)
- User documentation, examples, and workshop materials (4 months)

Of the approximately 50 person-months allocated in the budget, this plan accounts for 40 months—or 80% of total project time—devoted to specific development tasks. The remaining 20% is necessary to account for weekly planning and status meetings, general project management, dissemination efforts, and vacation/sick leave.

Key Personnel

The principal investigators bring to the project extensive experience in XML, web, and learning standards; development of mathematical software applications; science and mathematics publishing; project management; and commercial business operations. The project team will consult with a three-member advisory group that includes representatives from other MathDL projects, ensuring that techexplorer addresses the technical needs and commercialization efforts of MathDL.

Samuel S. Dooley, software architect, has been a senior member of the techexplorer development team since its early inception, and is the designer of techexplorer's equation editor. He will be responsible for redesigning the techexplorer plug-in architecture, including algorithms and API functionality; writing the architecture document and implementation plan; and ensuring compliance with XML and learning metadata standards.

Donald W. DeLand, project manager, has many years of experience in the production and management of STEM print and media publications, and as a technical writer. He will write the requirements document and quarterly status reports; monitor progress and adherence to requirements; and oversee interface design, user documentation, and dissemination. In addition, he will serve as the primary liaison to the advisory group, and will chair the meetings and conference calls between the development team and the advisory group.

Senior programmer (to be hired) will be responsible for day-to-day programming activities; adherence to the implementation plan; design of quality assurance testing; and resolving browser-specific implementation issues.

Warren Leach, staff programmer, brings to the project valuable experience in Macintosh and TeX software development. He will be responsible for implementing many of the changes to techexplorer's core rendering engine; code optimization; and porting to Macintosh operating systems.

Staff programmer (to be hired) will be responsible for porting to Windows and Linux variants; implementing user interfaces; tracking bug reports; and implementing the quality assurance testing.

The advisory group, which represents various activities of MathDL, will include the following:

Dr. Frank Wattenberg, Professor, Department of Mathematics, United States Military Academy, is a long-time proponent of digital libraries, former director of the Math Forum, and author of a forthcoming online textbook on mathematical modeling for the MathDL Books Online project.

Dr. David A. Smith, Associate Professor Emeritus, Department of Mathematics, Duke University, is the editor of MathDL's *Journal of Online Mathematics* (JOMA), co-director of the Connected Curriculum Project, and co-author, with Dr. Lawrence Moore, of a forthcoming online calculus textbook for the MathDL Books Online project.

Dr. Daniel H. Steinberg, Director of Java Offerings, Dim Sum Thinking, is adjunct professor of computer science at John Carroll University, the author of several books on Java programming and extreme programming, editor of JOMA's Mathlets library, and contributing editor of JOMA's Developer's Corner.

Timeline for techexplorer Development

- November 2003: Start of project; initial planning meeting of development team; planning meeting with advisory group at International Conference on Technology in Collegiate Mathematics (ICTCM)
- December 2003: First draft of Requirements and Architecture documents
- January 2004: Joint Mathematics Meeting; meeting with advisory group; first-quarter status report; prototypes for equation-editing GUIs
- April 2004: second-quarter status report; conference call with advisory group
- July 2004: third-quarter status report; conference call with advisory group
- August 2004: MAA MathFest workshop on authoring interactive mathematics; begin final implementation of Macintosh and Linux ports; begin final documentation and quality assurance testing

- October 2004: fourth-quarter status report; conference call with advisory group; final debugging; begin final report to NSF

Dissemination

The primary goal of this project is to provide MathDL users—instructors and their students—with technology that enables high-quality, web-based interactive mathematics. To this end, techexplorer will be freely available for individual download from the MathDL web site, together with extensive user documentation, frequently asked questions, and examples of "best practice" applications. The project team will also work proactively to market techexplorer's utility within and beyond MathDL and the NSDL through the following activities:

- Participating in MathDL, Math Forum, and MAA discussions and workshops that address interactive mathematics and web-based or computer-aided instruction.
- Promoting techexplorer's utility to other NSDL libraries through participation in the Mathematical Sciences Conference Group on Digital Educational Resources (composed of representatives from various digital libraries that have a shared interest in mathematics) and with a summary article in NSDL's *D-Lib* online journal.
- Submitting a minimum of two articles to MathDL's *Journal of Online Mathematics and Its Applications* (JOMA). These articles will draw on the experience of the development team outside of academia, and will address issues related to techexplorer development that do not fall under the category of "user documentation" but are of interest to JOMA readers. Topic selection will be made with the help of the advisory group, and will likely address issues such as the role and status of web standards in interactive mathematics, current implementations of web-math applications, and user interactions with web-math content.
- Submitting a minimum of six Java applets for inclusion in the JOMA mathlet library. Emphasis will be on mathlets that use techexplorer.
- Writing tutorial materials that can be used in workshops on creating web-based mathematics. Such workshops are sponsored by the MAA and MathDL, and are conducted as webcasts and in conjunction with the two annual MAA meetings (MathFest and the Joint Mathematics Meeting). The developers will participate in these workshops at the discretion of the workshop organizers.
- Submitting additional journal articles and presentations to non-NSDL forums that are particularly relevant to techexplorer development, including the International Symposium on Symbolic and Algebraic Computation (ISSAC) and the International Conference on MathML and Technologies for Math on the Web.
- Issuing press releases and arranging no-cost promotional announcements that advertise the availability of techexplorer on the MathDL web site, and discussing the collaboration between techexplorer development and MathDL as a model for cooperation among industry, education, and government. These announcements will be submitted to professional journals that target broad science, mathematics, and engineering audiences, such as *Science*, *Physics Today*, *IEEE Spectrum*, and *American Scientist*.

Evaluation

The development team will conduct rigorous quality assurance testing of techexplorer functionality throughout the course of development and particularly during the final three months of the project. The team will develop a series of benchmarks that measure techexplorer's performance; collect and organize a test suite of materials written for techexplorer in order to evaluate techexplorer APIs; and use the W3C MathML Working Group's extensive test suite of MathML code fragments, supplemented by numerous $\TeX$ examples from Integre's composition archives, to evaluate techexplorer's rendering engine and MathML support. Numerous examples have already been compiled by IBM's development team, and this material will provide a good basis for early-stage testing.

Close collaboration between the development team and the project's MathDL advisory board will ensure that techexplorer addresses the functionality and reliability demanded by the MathDL audience. Quarterly reports detailing the project's status, technical and usability issues, and MathDL concerns will be submitted to the advisory board and serve as an agenda for the subsequent quarterly meetings or conference calls.

Input and feedback from end users will be garnered through the active participation of the project developers in JOMA's Developers' Area, an online forum in which the MathDL audience discusses issues related to the creation and dissemination of interactive mathematics. In addition, the techexplorer plug-in will be submitted to the MathTools section of the Math Forum. MathTools includes its own Developers' Area, an online Roundtable Discussion, and user reviews of mathematics software. These forums will provide unbiased evaluations of techexplorer's utility to the mathematics community, and will serve to identify technical or usability flaws that need to be addressed.

Sustainability

For MathDL, techexplorer provides a launching point for delivery of subscription-based web services. A portion of subscription revenues for services that rely on techexplorer may be allocated to techexplorer maintenance and technical support. Our current plan is for the Math Forum to provide first-tier technical support for MathDL users, supplemented by extensive online documentation and examples, Frequently Asked Questions (FAQs), and usenet forums. Project developers will provide second-tier technical support via the Math Forum. These arrangements, as well as techexplorer's role in sustaining MathDL, will be determined in conjunction with the advisory group and project directors for MathDL and the Math Forum. In recognition of MathDL's important role in techexplorer development, and as a guarantee of Integre's commitment to MathDL's mission, Integre will grant to MathDL a perpetual, nonexclusive, nonrevokable, and nontransferable license to redistribute techexplorer. MathDL's success in demonstrating the benefits of techexplorer in mathematics education will attract commercial interest in techexplorer and in other MathDL projects.

Beyond MathDL, the techexplorer plug-in will be licensed by Integre to other organizations that would benefit from an ability to redistribute the plug-in. Identifying interested organizations will not be difficult: for three years Integre collaborated with IBM on techexplorer marketing, so we already have a strong sense of where that interest lies, what features those organizations are needing, and what the price point is for those licenses. Revenue from these redistribution licenses will constitute the primary financial support needed to sustain techexplorer development and technical support.

As the web evolves, there will be tremendous pressure on commercial, government, and educational organizations to participate in new business models, and to support semantic markup in their web content [7, 8]. For organizations that require mathematics, techexplorer will provide that "semantic web" support.