TechTALK: A Web Based System for Mathematical Collaboration

Lakshman Y. N.*
Department of Mathematics and Computer Science
Drexel University
Philadelphia, PA 19104, USA

Michael R. Miller
Siemens Corporate Research, Princeton, NJ 08536, USA

Santiago Lombeyda †
Department of Mathematics and Computer Science
Drexel University
Philadelphia, PA 19104, USA

April 22, 1999

1 Introduction

Imagine two mathematicians working at a blackboard, discussing their work as it develops; imagine three students sitting around a workstation, working on a homework assignment with the help of a mathematics system such as Maple or Matlab. What if one of the mathematicians is in Philadelphia, and the other is in Bombay? What if two of the students are in their dorm rooms while the third is at her parents' home? TechTalk is a mathematical collaboration system for the World Wide Web. It enables a group of people in different locations to perform calculations, generate graphical output, and discuss their results. It allows them to interact as though they were in the same room, even if they are located in different parts of the world. TechTalk attempts to simulate one or more rooms with several people with access to Maple and Matlab in each room

*Supported in part by NSF Grants CCR-9527130 and CCR-9712219; Current Address: Computing Sciences Research, Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974

†Current Address: Center for Advanced Computing Research, California Institute of Technology, 1200 East California Boulevard, Pasadena, CA 91125

(imagine all the people in a room looking at the same screen, but each with a separate keyboard). Since people sitting in a room tend to talk to each other, TechTalk includes support for conversation. A user needs only a Java–enabled web browser to participate in a TechTalk session.

2 Computing and Collaborating on the Internet

In the past several years, we have seen a maturing of general purpose computer algebra systems such as Maple and Mathematica and the evolution of several "next-generation" systems (Magma, Macaulay II, and Kant to name a few). Besides having extensive libraries of mathematical functions, these systems come with the capability to write generic programs. There are also numerous special purpose packages that are meant to perform huge computations in particular domains – for instance, PARI and Kant for number theory, Singular and GB for commutative algebra. Often, there is a need to use the capabilities of more than one of these systems/packages to solve a problem. For example, in order to solve a certain system of polynomial equations, one might perform some initial simplification of the equations using Maple, then compute a Gröbner basis for the ideal generated by the system on GB, use Maple again to generate matrices of multiplication for certain endomorphisms on the residue class ring modulo the ideal generated by the input polynomials and then use a numerical package to compute the eigenvalues of these matrices from which one can infer the solutions to the input polynomial system. We would like to be able to perform such computations involving several computational engines, possibly running on very different hardware configurations located in different geographical locations, in a seamless manner. With the wide spread use of the World-Wide-Web, we find code repositories and servers accessible from the web that provide mathematical services – for example, the Netlib facility offers a huge repository of routines for solving a host of numerical problems [**?**]; the NEOS server at the Argonne National Laboratories [**?**] offers numerical optimization services and the partial differential equations server at Purdue University [**?**] offers the capability of solving partial differential equations. We would like to be able to configure such servers in a smooth manner to carry out computations that might need multiple services.

In recent years, computer algebra systems have made deep inroads into the mathematical curricula at the high school and college levels. There has been much discussion on harnessing the power of computers for effective teaching of mathematics [**?**]. There appears to be general agreement that too much time may be spent in learning the intricacies of one or more computer algebra systems and timely answers to specific questions concerning the use of these systems may be of great help to students. We believe that a web-based tutorial environment can be quite helpful in this regard. On a certain level, such an environment would require infrastructure similar to the distributed mathematical

computations described earlier.

TechTalk was designed as a prototype to gain an understanding of the issues involved in the above situations and to perform some experiments in collaborative computing and web- based tutorials for elementary mathematics. There have been servers offering mathematical services, including Maple and Mathematica servers, on the web. Most of these offer a single user the ability to interact with a server. TechTalk offers a much richer environment where multiple users can share multiple mathematical servers, chain computations on different servers, talk to each other, record and replay sessions etc. A user enters a TechTalk-session (or TechTalk-room) by hitting a web-site that is set up for running TechTalk. A password mechanism ensures that only authorized users can enter the room. Upon entering a TechTalk room, a user can interact with Maple, Matlab and other users in the room. TechTalk offers the following features:

- TechTalk makes it possible for multiple users in different locations to share Maple and Matlab sessions running on remote servers. Each user can send Maple/Matlab commands/programs to the remote servers. The results of the computations can be viewed by all the users in a TechTalk room or just by the sender of the command. The results of a computation can be textual/graphic.

- Under certain situations, the results of a Maple computation can be sent directly to the Matlab server and the results of the Matlab computation observed by all the users: for instance, a user might use Maple to generate Matlab code for solving an ordinary differential equation and send the code directly to the Matlab server and have its output (a color plot, perhaps) displayed to all the users.

- The users can have a multi-way conversation in a separate talk window on each user's screen. A logging facility enables one to save a session and replay either the entire session or selected portions of it; the flexible replay facility allows different uses of a log including broadcasting of class lessons/notes in an electronic tutorial setting.

- TechTalk is implemented in the Java programming language. This has several obvious advantages for the designer and the users of the system. TechTalk takes advantage of the communication infrastructure offered by the world-wide-web. Java makes it easier to use this infrastructure. The implementation is largely platform independent. Users of TechTalk only require a Java-enabled web-browser; users of TechTalk do not need to have accounts on the server/s running TechTalk.

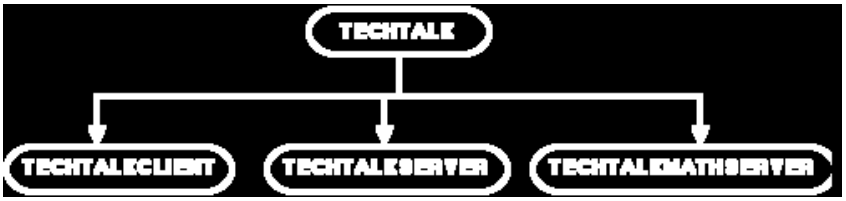
In our set-up, the Maple and Matlab servers run on Unix systems and the users of TechTalk connect from Unix/Windows/Macintosh systems. After a

simple, guided initial set-up, TechTalk requires very little additional administration; such things as adding new users or changing the host servers can be done remotely over the web by an authorized administrator. TechTalk has been successfully used to conduct remote tutorial sessions in an elementary numerical analysis course. It has also been used to conduct mathematical conversations (involving computations) among small groups of people in different geographic locations. In the rest of this paper, we describe TechTalk system and its architecture in detail and mention some of the experiments that we have conducted with it. More details of the use of TechTalk as a web-based tutorial for teaching of mathematics can be found in [?].

3 TechTalk Behind the Scenes

In this section we give a short description of the architecture of TechTalk. There are three main components to TechTalk: the TechTalkClient, the TechTalkServer and the TechMathServers.

- The TechTalkClient handles client side affairs such as maintaining displays, packaging client requests and interpreting server responses.
- The TechTalkServer holds together a TechTalk session by maintaining connections to and relevant information about the users in a session. It also maintains connections to any number of mathematical engines such as Maple/Matlab (in principle, TechTalk can wrap any interactive program as a server; we use Maple and Matlab as they were suitable for our purposes and easily available to us). The TechTalkServer collects client requests, interprets and routes them to the appropriate mathematical engines. It also directs the responses from the mathematical engines to the appropriate clients waiting for those responses.
- The TechMathServers are essentially wrapper programs that make servers out of interactive programs such as Maple and Matlab. One TechMathServer is needed for each mathematical server. It maintains a connection with a TechTalkServer, interprets computational requests (which are commands that the mathematical engine can understand, wrapped in different protocols), passes them to the mathematical engine, and sends responses from the engine to the TechTalkServer after appropriate packaging.



We now describe the capabilities of each of these components in detail.

3.1 The TechTalkClient

The TechTalkClient is responsible for handling client-side affairs such as managing the displays and relaying user inputs/responses to the servers. This component is implemented as a Java applet. When a user hits a TechTalk session from a web browser, the applet comes over to his machine and starts communicating with a TechTalkServer. The TechTalkClient performs the following functions:

- Requests user id and password for authentication by the TechTalkServer. It opens up a communication link with the TechTalkServer running on the same host that it (the applet) came from and passes the information collected from the user to the TechTalkServer in a convenient form.

- Once a user is authenticated by the TechTalkServer, the TechTalkClient applet opens up a window that can be used to communicate with Maple/Matlab or with other users in the session. A user can direct a message to any of the above destinations by hitting buttons labelled as "to Maple", "to Matlab". Alternately, a user can request three separate windows – a Maple window, a Matlab window and a conversation window – for communicating with the servers and other users. The windows have several buttons for sending commands to the servers and getting various kinds of information. For instance, in the conversation window, there is a "list users" button that causes the client to obtain a list of all users currently logged on to the TechTalk session and display it. In the Maple and Matlab windows, the client applet tries to present interfaces close to the standard Maple/Matlab interfaces respectively. The current implementation supports two- dimensional textual output from Maple and Matlab. The user can type text into any window. The client applet packages chunks of text from a window with additional information such as the name of the window the text was typed into, whether the user requested the results to be broadcast/returned only to him, and the length of the text. It then sends the package to the TechTalkServer.

- The client applet receives information from the TechTalkServer in packages with information about what to do with it, such as which window to display it in, and displays it accordingly. The client applet does not reformat textual output from Maple/Matlab in the present implementation.

- A user with permission to do so can login as an administrator.An administrator performs such functions as maintaining a list of all bona fide users and their passwords, creating/terminating users, fixing the numbers and locations of various servers etc. If a user logs in as an administrator, the client applet displays a special button (in addition to the various windows) called the Administrator Button. Pressing this button causes the client to show an editor interface for changing any of the above information. At any time, only one user can login as administrator to ensure integrity of

administrative information. Password files and information about server locations and socket numbers are maintained on the same file system as the one used by the host running the TechTalkServer. Any request from an administrator to update this information is handled by the TechTalk-Server.

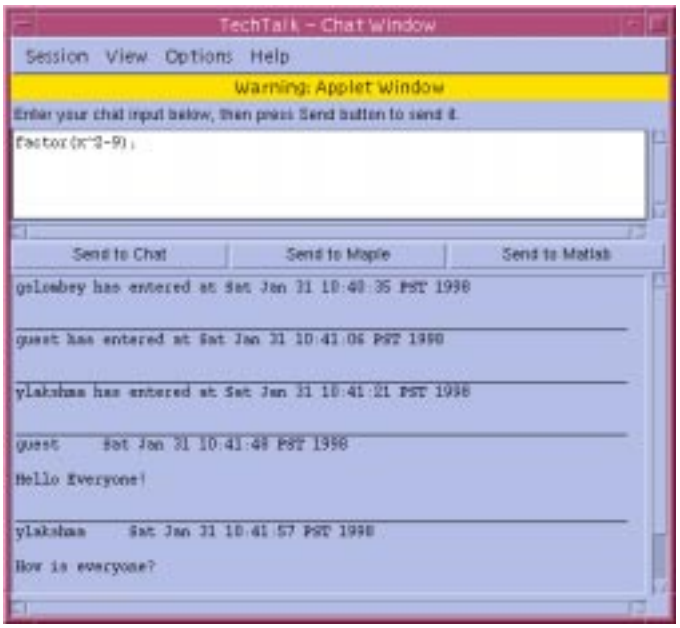
- A user can choose to record an entire TechTalk session (or a part of it) in a log or peruse the log of an earlier session. The StartLog, EndLog and ShowLog buttons enable the user to perform these tasks. The log facility is particularly useful and we will describe it in more detail later.

3.2 The TechTalkServer

The TechTalkServer is the hub of the activities performed by TechTalk. The TechTalkServer is a stand-alone Java application that must be running on the same host as the web-server that houses the TechTalk homepage and hands out the TechTalkClient applet. The TechTalkServer maintains connections to the TechTalkClient applets for every user in each TechTalk room as well as various servers offering Maple/Matlab services and directs the flow of information among these. As part of the system set-up, one has to start up the TechTalkServer and several Maple/Matlab servers. All this is achieved by a single command that uses information about the number, type and locations of the servers; this information is to be maintained by the person responsible

for running TechTalk. During the start-up process, the TechTalkServer establishes connections to the Maple/Matlab servers and maintains those connections through the life of the TechTalk session. When a TechTalk user downloads the TechTalkClient applet from the web-site offering TechTalk, the applet connects to the TechTalkServer running on the same host as the one on which the TechTalk web-page resides. A single TechTalkServer is capable of supporting several TechTalk rooms/sessions. Some of the specific functions that the TechTalkServer performs are:

- **Password Authentication:** TechTalkServer is responsible for authenticating TechTalk users. It has access to a password file which it uses to authenticate the user id- password pair sent to it by TechTalkClient applets. The TechTalkServer recognizes two kinds of users: administrators and regular users. The type of a user is part of the information maintained in the password files. Password information is transmitted after encryption by the TechTalkClient. The TechTalkServer maintains a list of active users for each TechTalk room that it supports. After a user is authenticated, the TechTalkServer adds the user name and the location of the user to the list of active users for the particular TechTalk room that the user has chosen.



- **Conversation Handling:** Users in a TechTalk room converse by typing text into his conversation window and pressing a "send" button. The user's TechTalkClient applet packages the text along with information

about the user's identity, the room to which the user belongs and the type of information – conversational text in this case, and sends it to a specific port on the host running the TechTalkServer on which the TechTalkServer is listening. A listener thread constantly monitors the port. Upon receiving such a package, the listener thread hands it off to another thread for dealing with it and goes back to its monitoring duty. The other thread extracts the text and sends it to every user on the current list of active users who are in the same TechTalk room as the sender. Other requests from the conversation window such as for listing all the users in a TechTalk room are handled similarly (this request results in a response only to the client that requested it).

- **Maple/Matlab Requests:** A user can type any Maple/Matlab commands into the appropriate window and press the send button. The commands can be multi-line/single line. The results of a command can be either broadcast to all the users in a room or sent back just to the person issuing the command (we have found the second mode to be helpful in somewhat embarrassing situations such as when we have forgotten the first three decimal digits of some universal constants). The user can specify the mode by ticking off a toggle on the appropriate window. As before, the user's TechTalkClient applet packages the command along with information about the user's identity, the room to which the user belongs and the type of information – Maple/Matlab commands in this case and the response mode (broadcast/single user response) sends it to the listening port of the TechTalkServer. The client does not perform any syntax check –this is done by the actual mathematical system itself and if there are syntactic errors, the user receives error messages/warnings from the particular system (just as when the user types commands directly into Maple/Matlab). On receiving a package containing user commands, the listener thread hands it off to another thread. This thread creates a new package containing the commands and the source of the command (the user's identity and location), whether the results are to be broadcast or not, and sends it to the appropriate Maple/Matlab server. Each TechTalk room has a separate Maple and Matlab server at this point. All users in a TechTalk room interact with them. Certain shell escapes from Maple/Matlab are disabled (a user cannot issue them from the applet) for security reasons. The responses from Maple/Matlab are sent in packages (created by TechMathServer). They contain the actual response from the Maple/Matlab session to a command sequence and some header information indicating the source of the response (Maple/Matlab) and the intended recipient (user name, TechTalk room number, broadcast/single–user response). The TechTalkServer handles the package based on the header information.

- **Graphical Output:** Graphical output from Maple/Matlab is handled differently. Maple/Matlab is made to save the graphical output in GIF format and send the location of the file containing the output. The TechTalkServer conveys this location to the TechTalkClient applets that are supposed to see the output which then load the image on to the users' screens. This mechanism requires the GIF files to be saved on the same host as the TechTalkServer since the TechTalkClient applets can only connect to that one host. In our present implementation, we run the Maple/Matlab servers on different hosts that share the same file system as the one running the TechTalkServer. The GIF files are saved in a directory known to all the servers and no special protocols are needed for this task. Maple to Matlab: This mode lets a user type commands in to a Maple window and have the results sent directly to a Matlab server and the results of the Matlab computation displayed to the TechTalk room. At this time, it is assumed that the output of the Maple command/s will be understood by Matlab. We have used this mode to generate Matlab code for solving ODEs and PDEs in Maple and send the code directly to Matlab and then display the Matlab output (usually graphical) to a TechTalk room. For the TechTalkServer, this is a two step process, the first one involving an interaction with a Maple server followed by an interaction with a Matlab server. Messages in this mode are tagged differently from others and trigger different protocols.

- **Recording a TechTalk session:** A user can choose to record a TechTalk session by pressing the StartLog button. Upon receiving a request to record a session, the TechTalkServer issues a message to all the users currently in the same room as the requesting user to the effect that the current session is being recorded. Since the typical TechTalk session has a small number of cooperating participants, we assume that such requests are agreed upon by all the participants in the room. It prompts the user for an identifying session name, whether the user wants only the Maple part, or Matlab part, or the chat part, or all of the session recorded. Messages coming from a client who is in the "record session" mode are specially tagged and depending on what parts of the session the user wants saved, the TechTalkServer saves the relevant messages passing through it into a named file after putting some tags on them. Allowing clients to save sessions on the host server introduces some administrative problems such as space management and there is also the potential for some user to inadvertently/maliciously fill up the host's disk. To prevent this, the TechTalk administrator can set limits on the amount of space available to a single user for logging purposes. It also helps to clear up logs older than a certain length of time. Pressing the end-log button ends the recording (the session continues).

- **Replaying a log:** A user can view the log of an earlier session by pressing the ShowLog button. He will be prompted for the session name (name of the file in which the log is stored). After checking access rights, the TechTalkServer will ask the user whether he wants to view the entire log or only portions of it (such as just the Maple queries and responses in the log). The log file is shown in a separate window with basic scrolling and the capability to jump to the next/previous (next + a number, previous - a number) chunk of information. A chunk of information is one entire query, or the response to a query, or a single chat message. The user can cut and paste information from the log-window to any of the other windows.

Each type of interaction described here is implemented through a general set of protocols that can also be used in other settings. The TechTalkServer clearly handles a lot of traffic and can be a bottle-neck. In our trials with up to a couple of dozens of users, we have not experienced any significant delay other than web-traffic delays (which is location and time dependent).

3.3 The TechMathServer

The TechMathServer module is essentially a Java wrapper for stand-alone applications such as Maple and Matlab. It communicates with the TechTalkServer on one side and an application (Maple or Matlab in this case) on the other.

The TechMathServer controls the standard input/output/error streams of the application it is running. It receives requests from the TechTalkServer (which is one of the packages described earlier), extracts the commands to be sent to the application, sends the commands to the application and waits. Requests that arrive while the TechMathServer is waiting are queued on a first-come-first-served basis.

The TechMathServer has special ways of handling input errors, unexpected command termination etc. and these depend largely on the application at this time since there are no standard ways yet for applications such as Maple to interact with other programs in a client- server mode. TechMathServer also performs some buffering since the size of the output in many cases in unpredictable and can be quite large.

The TechMathServer is fairly general in its conception and implementation in the sense that it can be used as a wrapper for other applications besides Maple/Matlab. Still, it is the most application and platform dependent part of the TechTalk system. The Java language makes the platform dependence less of a problem than in other languages. We are currently investigating the use of common standards such as MathLink and OpenMath for exchanging mathematical information to decrease application specific actions in TechMathServer.

3.4 System Start-Up and Shut-Down

Starting up a system like TechTalk with multiple servers on different hosts can be quite a task. Recognizing this, we have made up a command to start up TechTalk and a command to shut down TechTalk. These commands make use of some initialization files and the password file. The password file contains basic user information, the privileges that each user has (guest, regular, administrator) and encrypted passwords.

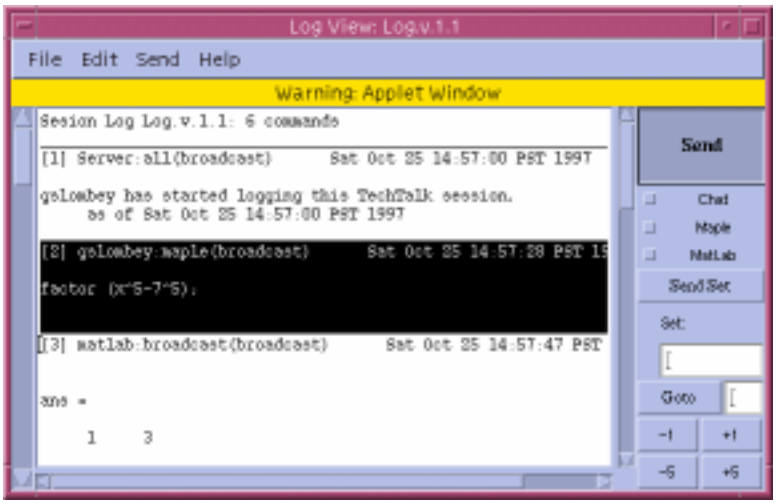
The initialization files contain such information as the hostnames of the hosts running the various servers along with relevant port numbers, fully qualified path names of directories to be used for temporarily storing GIF files and log files and access permissions for various user levels. These files have to be created before starting TechTalk.

A stand-alone initialization program called InitTT prompts the administrator for all the information needed to start TechTalk and creates the password and initialization files. The StartTT command starts a TechTalkServer and several TechMathServer-s running Maple/Matlab (their number and resident host information is found in the initialization files). The ShutTT command needs a handle to a TechTalk session and systematically shuts down all the servers that are part of the TechTalk session with the given handle. It also removes several temporary files/GIF files created by the TechTalk session. The ClearTT command can be invoked periodically to delete dated log-files and GIF files to free up space on the host running the TechTalkServer. The current implementation of these commands assumes that all the TechTalk servers are running on a network of computers with a shared file system. The StartTT, ShutTT and ClearTT commands are written in Perl. A short online manual provides all the information needed to set up and use TechTalk [**?**].

3.5 The Log Facility

As mentioned before, users can log part or all of a session and replay it later. The TechTalkServer views a log file as a sequence of messages where each message is one unit of information that passed through the TechTalkServer at some point. The messages are tagged with information about the source, recipient/s and other information. A user can replay a session by hitting the ShowLog button. The user is shown a list of viewable session logs and asked to choose one. The user can only choose to view a session that he was part of. The user can choose to view the entire session, or just the Maple queries and responses (including graphical output), or just the Matlab queries and responses or just the chat part or some combination of these. The chosen part of the log is displayed in a separate window. The displayed messages are numbered and can be referred to by those numbers. The log window has basic scroll features and the user can display messages numbered next + increment/previous - decrement or a range of messages. The user can reissue a command or a range of commands from the

log window by either referring to them by their numbers or cutting and pasting the commands to TechTalk's Maple/Matlab window. The results are displayed in the Maple/Matlab window as appropriate. Simple editing features enable the user to make changes to a command from the log file prior to reissuing it.



The log files are stored as HTML files (with some additional tags) and can be viewed independently if desired. Allowing TechTalk users to save logs introduces the problem of maintaining files and the risk of someone accidentally or maliciously filling up TechTalk's host's disk space. We adopt a simple policy of allowing each user a certain maximum amount of disk space (to which the user has access only through the TechTalkServer) and log files have a fixed lifetime after which they are deleted. A log file can only be viewed by the creator of the log file or in his company.

The log facility turns out to be particularly helpful when TechTalk is used in a classroom/tutorial setting. For example, an instructor can prepare a Maple lesson, test it out in a TechTalk session, save it as a log and replay it to the class later, or a student can study the lesson on his own at a later time. Students can save their work and show it to a tutor who can see exactly what the students did.

TechTalk was used in a small class on introductory numerical analysis without the log facility. We plan to use it (with the log facility) in a large introductory calculus class in the near future. Our experiences with using TechTalk in the introductory linear algebra class are reported in [**?**].

4 Lessons Learnt and Future Plans

In implementing TechTalk, we ran into two kinds of problems: the first was that the mathematical packages that we used were not built to be used as servers;

the second kind had to do with the communication support provided by the internet. One of the key features of a TechTalk-like system is that mathematical information can be passed from one engine to the other. To do this in a reasonable way, the various engines should share a common language or interface. At present, there is no adequate candidate for this job. In the context of computer algebra, open standards are under development for exchanging mathematical information. The OpenMath[?] standard proposed by the OpenMath consortium focusses on semantic markup and the MathML[?] standard proposed by the World Wide Wed Consortium's task group for mathematics concentrates on presentation mark-up. They provide a common notation for sharing information involving the basic entities that one plays with in computer algebra systems (polynomials, rational functions, general arithmetic expressions) and the ability to extend this notation to other composite types. While these may be adequate in a situation involving the interaction of one or more pure symbolic systems, we need the ability to represent a much richer domain of entities as we are interested in setting up interactions among engines that perform quite different types of computation – for instance, a symbolic, a numeric and a visualization engine. In our current set-up, we pass Matlab code or just numerical matrices from Maple to Matlab. If we were to use say, the NAG Explorer for visualization, then, we would need to provide a "lattice" for the visualizer. It is essential to bring together the efforts of the symbolic, numeric and visualization communities to evolve acceptable standards for communicating mathematical information.

As users of software systems, we have to get used to different interfaces on different machines. Such a situation in an introductory class (using two different interfaces for Maple, for example) can create tremendous confusion. This is a problem we have had to face in building TechTalk. The user interfaces provided by systems such as Mathematica and Maple are getting more sophisticated with each new release; the parts of the system that handle the interface are proprietary and built into the system (meaning, one can not find a free floating module that can recreate the Mathematica interface). The effort involved in replicating the interfaces of all the systems that we might want to connect in a TechTalk session is tremendous. Again, it would be useful to have these systems put out things for display in a standard mark-up language. There are efforts underway to evolve such standards, notably, the MathML working group under the WWW consortium is attempting to come up with specifications for a mark-up language for mathematical notation [?].

Another difficulty encountered in building TechTalk was the process of coaxing Maple and Matlab to pretend to be servers. Considerable effort had to be put in to understand how these systems handle low level I/O. The task was made harder by the fact that these details are not spelt out and are very hard to get out of the makers of the systems. An even more vexing problem was that having found out some of the intricacies of a particular system, we found that a new release of the system had quite different ways of doing I/O! All this of course points to a need for acceptable server interfaces. In light of the problems

mentioned with replicating interfaces, we feel that the nice interfaces that some of these systems provide may be in the way when it comes to their functioning as servers. Maple provides a version called MathEdge that comes without the worksheet interface but all the other parts of Maple intact. Such a stripped down version capable of communicating through a standard mark-up language with the rest of the world would be a desirable thing. In addition, the server version ought to be able to respond to simple remote system queries/commands such as "are you ready for the next job?", "are you busy?", "please stop whatever you are doing!" The need for platform independence cannot be over-emphasized. Based on our experiences with building these servers, we are currently developing a generic "server for mathematics" to work with existing computational programs inlcuding typical interactive mathematics systems (IMS). The design of our generic server is guided by our desire to

- provide simple mechanisms to encapsulate commercial or non-commercial mathematics engines (such as Maple, Matlab, Mathematica, Singular, Gb or other Gröbner software[?] etc.), as well as suitably wrapped up code libraries for doing symbolic, numerical and graphic processing into servers capable of providing their services to client requests on the Internet;
- provide means for incorporating these mathematical servers as components of distributed applications including a simple and powerful communication infrastructure;
- support existing and upcoming protocols for exchanging mathematical/graphical information;
- use existing commercial and public domain software as much as possible to realize the above goals.

More details of the generic server can be found in [?].

In using TechTalk locally, we found hardly any delays in communication. While using it long distance (coast-to-coast), occasionally, we found serious network delays. One of the users would not see anything for a long time and a bunch of things all at once. The delays are not uniform and appear to be location and time dependent. In any case, this is not acceptable for a collaboration tool! Currently, the amount of data sent between the various servers and clients is not large (a few kilobyte chunks) and we are trying to understand the reasons for the delays and what can be done to lessen them.

Currently, we are working on a new version of TechTalk. Our plans include the use of OpenMath with some enhancements to set up communication protocols among symbolic and numeric engines. We also plan to provide a visual interface for specifying and setting up distributed computations and the ability to access code repositories on the web. The Java bytecode for the current implementation of TechTalk along with instructions for installation and use can be downloaded from `http://penguin.mcs.drexel.edu`, see [?].

Acknowledgements

The implementation of TechTalk's administrative module was performed by Hoi-Man Chang and Todd Zino. We thank our PSE group colleagues at Drexel, Bruce Char, Tom Hewett, Jeremy Johnson and Raji Venkatesan for many helpful discussions and Joseph Kiniry of CalTech for detailed comments on installing TechTalk. Our thanks to Randy Bramley of the University of Indiana at Bloominton for editorial comments that greatly improved this presentation.

@InProceedingsavitzur95, author = Ron Avitzur and Olaf Bachmann and Norbert Kajler, title = From Honest to Intelligent Plotting, booktitle = Proceedings of the 1995 International Symposium on Symbolic and Algebraic Computation, editor = A.H.M. Levelt, year = 1995, publisher = ACM Press, pages = 32-41

@InProceedingsbayram92, author = M. Bayram and J.P. Bennett, title = A simple general purpose technique for interfacing between Computer Algebra and numerical analysis systems, booktitle = International Symposium on the design and implementation of symbolic computation systems, DISCO'90, editor = John Fitch, year = 1992, publisher = Springer-Verlag, pages = 145-151, note = Lecture Notes in Computer Science 721, annote = Implements Fortran-Reduce interprocess communication.

@InProceedingscentralControl, author = Stéphane Dalmas and Marc Gaëtano, title = Making systems communicate and cooperate: the central control approach, booktitle = International Symposium on the design and implementation of symbolic computation systems, DISCO'96, editor = Jacques Calmet and Carla Limongelli, year = 1996, publisher = Springer-Verlag, pages = 308-319

@InProceedingschar89, author = Bruce W. Char, title = Automatic Reasoning About Numerical Stability of Rational Expressions, booktitle = Proceedings of the 1989 International Symposium on Symbolic and Algebraic Computation, editor = Gaston Gonnet, year = 1989, publisher = ACM Press, pages = 234-241

@Articledabdub95, author = Donald Dabdub and J.H. Seinfeld, title = Extrapolation techniques used in the solution of stiff ODEs associated with chemical kinetics of air quality models, journal = Atmospheric Environment, year = 1995, volume = 29, pages = 403-410,

@Articledabdub97, author = A. Sandu and J.G. Verwer and M. van Loon and others, title = Benchmarking stiff ODE solvers for atmospheric chemistry problems I: implicit versus explicit, journal = Atmospheric Environment, year = 1975, pages = 3149-3164

@Articledabdub98, author = D. Dabdub and R. Manohar, title = Performance and portability of an air quality model, journal = Parallel Computing, year = 1998, note = To appear.

@MiscdabdubPersonal, author = Donald Dabdub, title = Personal Communication, year = 1998, month = January

@InProceedingsdsc91, author = A. Diaz and E. Kaltofen and K. Schmitz and T. Valente, title = DSC: A system for distributed symbolic computation,

booktitle = Proceedings of the 1991 International Symposium on Symbolic and Algebraic Computation, editor = Stephen M. Watt, year = 1991, publisher = ACM Press, pages = 323-332

@Miscgenericserver, author = Y. N. Lakshman and Bruce Char and Jeremy Johnson, title = Software components using symbolic computation for problem solving environments, howpublished = manuscript, submitted for publication, year = 1998

@Miscgenserver, author = Y. N. Lakshman and Bruce Char, title = Implementing mathematical servers using a generic Java server: A user manual, howpublished = manuscript, under preparation, year = 1997

@InProceedingsinfo96, author = K. Mani Chandy and Adam Rifkin and Paolo A.G. Sivilotti and Jacob Mandelson and Matthew Richardson and Wesley Tanaka and Luke Weisman, title = A World-Wide Distributed System Using Java and the Internet., booktitle = IEEE International Symposium on High Performance Distributed Computing (HPDC-5), year = 1996, month = August, note = Available from `http:// www.infospheres.caltech.edu/ papers/ index.html`

@Articleinfo97, author = K. Mani Chandy and Joe Kiniry and Adam Rifkin and Dan Zimmerman, title = Webs of Archived Distributed Computations for Asynchronous Collaboration, journal = Journal of Supercomputing, year = 1997, volume = 11, number = 3, note = Available from `http:// www.infospheres.caltech.edu/ papers/ index.html`.

@Manualinfo97guide, title = Infospheres Infrastructure Documentation - APIs and User's Guide, author = K. Mani Chandy and Joe Kiniry and Adam Rifkin and Dan Zimmerman, organization = Infospheres Group, address = Department of Computer Science, California Institute of Technology, year = 1997, month = August, note = Available from `http:// www.infospheres.caltech.edu/ papers/ index.html`.

@Articleinfo98, author = K. Mani Chandy and Joe Kiniry and Adam Rifkin and Dan Zimmerman, title = A Framework for Structured Distributed Object Computing, journal = Parallel Computing, year = 1998, note = To appear. Available from `http: www.infospheres. caltech. edu/papers/index.html`.

@Miscinfo98a, key = info98a, title = The Caltech Infospheres Project, howpublished = OMG-DARPA-MCC Workshop on Compositional Software Architecture, year = 1998, month = January , note = Available at `http:// www.infospheres.caltech.edu/ infospheres.html`,

@ManualirisExplorerReference, title = IRIS Explorer version 3.5 Reference pages (Module and programmer's interface information), Unix edition, organization = Numerical Algorithms Group, Ltd., note = Available at `http:// www.nag.co.uk/visual/IE/ iecbb/ DOC/REF/ Index.html`

@ManualirisExplorerUsers, title = IRIS Explorer version 3.0 User's Guide, Unix version, organization = Numerical Algorithms Group, Ltd., OPTnote = Available at `http:// www.nag.co.uk/visual/ IE/iecbb/ DOC/UG/ CONTENTS.html`

@InProceedingsisabelle95, author = Clemens Ballarin and Karsten Homann and Jacques Calmet, title = Theorems and Algorithms: an interface between Isabelle and Maple, booktitle = Proceedings of the 1995 International Symposium on Symbolic and Algebraic Computation, editor = A.H.M. Levelt, year = 1995, publisher = ACM Press, pages = 150-157

@Bookjava97, author = David Flanagan, title = Java in a nutshell, publisher = O'Reilly, year = 1997, edition = Second

@InProceedingsKajler92, author = Norbert Kajler, title = Building a Computer Algebra Environment by Composition of Collaborative Tools, booktitle = Design and Implementation of Symbolic Computational Systems, Proceedings of DISCO '92, editor = John Fitch, year = 1992, publisher = Springer-Verlag, pages = 85-94, note = Lecture Notes in Computer Science 721

@InProceedingsKajler93, author = Robet Fournier and Norbert Kajler and Bernard Mourrain, title = IZIC: a portable language-driven tool for mathematical surfaces visualization, booktitle = International Symposium on the design and implementaiton of symbolic computation systems, DISCO'93, editor = Alfonso Miola, year = 1993, publisher = Springer-Verlag, pages = 341-353

@Bookjacobson92, author = Ivar Jacobson, title = Object-oriented Software engineering, publisher = ACM Press/Addison-Wesley, year = 1992

@MiscjavaStudio, key = Java Studio, title = Java Studio, howpublished = `http://www.sun.com/studio/index.html`, year = 1997

@Articlejscspec, author = L.Lambe(Ed.), title = Special Issue on Applications of Symbolic Computation to Research and Education, journal = Journal of Symbolic Computation, volume = 23, number = 5 and 6, year = 1997, month = May-June,

@Articlelsoda, author = Linda Petzold, title = Automatic Selection of Methods for Solving Stiff and Nonstiff Systems of Ordinary Differential Equations, journal = SIAM Journal of Scientific and Statistical Computing, year = 1983, volume = 4, number = 1, month = March

@TechReportmathlink, author = Wolfram Research Inc., title = MathLink Reference Guide (Mathematica Technical Report), institution = WRI, Inc., year = 1993

@Miscmathml, author = The MathML Working Group, title = Mathematical Markup Language, W3C Working Draft, howpublished = `http://www.w3.org/TR/WD-math`, year = 1998

@InProceedingsmp94, author = Simon Gray and Norbert Kajler and Paul Wang, title = MP: a protocol for efficient exchange of mathematical expressions, booktitle = Proceedings of the 1994 International Symposium on Symbolic and Algebraic Computation, year = 1994, pages = 330-335

@InProceedingsmp96, author = Simon Gray and Norbert Kajler and Paul Wang, title = Pluggability issues in the Multi Protocol, booktitle = International Symposium on the design and implementaiton of symbolic computation systems, DISCO'96, editor = Jacques Calmet and Carla Limongelli, year =

1996, publisher = Springer-Verlag, pages = 343-355, note = Lecture Notes in Computer Science 1128

@Miscneos, author = Neos Server Group, title = Network Enabled Optimization System, howpublished = http://www.mcs.anl.gov/home/otc/Server,

@Miscnetlib, author = Netlib Support, title = Netlib: A repository for mathematical software, data, documents, address lists and other useful things, howpublished = http://cm.bell-labs.com/netlib/,

@Booknierstrasz, author = Oscar Nierstrasz and Dennis Tsichritzis, title = Object-oriented software composition , publisher = Prentice Hall, year = 1995

@Miscopenmath, author = The OpenMath Consortium, title = The OpenMath Specification, Version 1.1, howpublished = http://www.openmath.org, year = 1997

@Bookorfali96, author = Robert Orfali and Dan harkey and Jeri Edwards, title = The Essential Distributed Objects Survival Guide, publisher = Wiley, year = 1996

@Articleperline91, author = Ron Perline and Joel Langer, title = Poisson geometry of the filament equation, journal = Journal of Nonlinear Science, year = 1, volume = 1, pages = 71-93

@Articleperline94, author = Ron Perline and Joel Langer, title = Local geometric invariants of integrable evolution equations, journal = Journal of Mathematical Physics, year = 1994, volume = 35, number = 4, pages = 1732-37

@manualpolymathBeans, organization = PolyMath Group, title = Description of Java OpenMath Demonstration Beans Version 0.3 (Preview Release), note = Available on the World Wide Web at http:// pdg.cecm.sfu.ca/openmath/lib/ bean.html, year = 1997

@Articlepsedefn, author = E. Gallopoulos and E.N. Houstis and J.R. Rice, title = Computer as Thinker/Doer: Problem-solving environments for Computational Science, journal = IEEE Computational Science and Engineering, volume = 1, number = 1, year = 1994, month = Summer, pages = 11-23

@Miscpurduepde, author = E.N.Houstis and J.R.Rice and S.Weerawarana and A.C.Catlin and P.Papachiou and K.-Y.Wang and M.Gaitatzes, title = Parallel (//) ELLPACK: A problem solving environment for pde based applications on multicomputer platforms, howpublished = ELLPACK Research Report, http://www.cs.purdue.edu/research/cse/pellpack/pellpack-paper-1.html

@InProceedingssui90, author = Y. Doleh and P.S. Wang, title = SUI: A system independent user interface for an integrated scientific computing environment, booktitle = Proceedings of the 1994 International Symposium on Symbolic and Algebraic Computation, year = 1990, pages = 88-95

@Bookszyperski, author = Clemens Szyperski, title = Component Software: Beyond Object-oriented programming, publisher = Addison-Wesley, year = 1997

@Miscteachtalk, author = J.R.Johnson and Y.N.Lakshman and P.Morgovsky and T.T.Hewett and T.Souder and S.Donegan and T.Fitzgerald, title = Virtual office hours in mathematics instruction, howpublished = manuscript, submitted for publication, year = 1998

@Miscttdesc, author = Y. N. Lakshman and Michael R. Miller and Santiago Lombeyda, title = TechTalk: A web based system for mathematical collaboration, howpublished = manuscript, submitted for publication, year = 1997

@Miscttmanual, author = Y. N. Lakshman and Santiago Lombeyda, title = TechTalk 1.0: User and Installation guide, howpublished = available from `http://penguin.mcs.drexel.edu/~techtalk`, year = 1997

@InProceedingswang97, author = Paul S. Wang, title = Tools for Parallel/Distributed Mathematical Computation, booktitle = Proceedings of the Second International Symposium on Parallel Symbolic Computation, PASCO '97, editor = Markus Hitz and Erich Kaltofen, year = 1997, publisher = ACM Press, pages = 188-195

@Unpublishedweber98, author = Andreas Weber and Wolfgang Küchlin and Bernhard Eggers and Volker Simonis, title = A parallel Gröbner solver as a web component, note = To be presented in ACM 1998 workshop on Java on high-performance network computing, February 28-March 1, 1998

@Miscxml, author = Tim Bray and Jean Paoli and C.M. Sperberg-McQueen, title = Extensible Markup Language (XML), Proposed W3C Recommendation PR-xml-971208, Dec 1997, howpublished = `http://www.w3.org/TR/PR-xml;`, year = 1997