Data Exchange and Telecollaboration: Technology in Support of New Models of Education

Alan Feldman <alan_feldman@terc.edu>
Lisa Johnson <lisa_johnson@terc.edu>
Daniel Lieberman <daniel_lieberman@terc.edu>
Irene Allen <irene_allen@terc.edu>
Johan van der Hoeven <johan_van_der_hoeven@terc.edu>

Abstract

The Testbed for Telecollaboration is a three-year initiative funded by the National Science Foundation to research and develop effective methods to support collaborative investigations and learning in communities of K–12 classrooms. Data exchange is a key concept of the Testbed. This paper describes the approach to data exchange that was first tried in the Testbed communities and some of the limitations that were encountered, and presents a direction for a far more flexible model. This new model simplifies data exchange among remote sites, facilitates the use of this data with a variety of data analysis tools, and gives users important support to guide their investigations and collaborations. Central to this approach is a new format for data exchange, which is outlined. We propose steps to define further the format and to create a widely used standard.

1 Introduction

The best uses of new technologies are not always obvious nor the first employed. "Steam sailboats"—conventional sailboats equipped with a steam engine—were an early application of steam power, designed to motor a sailboat out of doldrums. The inventors at the cutting edge at that time could not see that steam would lead to larger, all-steel boats that would use only steam and not have sails. Solving the problem of being stuck in the doldrums was useful, but because it was limited to the old paradigm of sailboats, the solution did not make very good use of the steam engine. In fact, in a short time the new paradigm rendered the steam sailboat itself irrelevant.

Many K–12 educational uses of networking are steam sailboats—slightly better ways of doing what educators have always done, but within the old paradigm. On-line encyclopedias and lectures-by-video are applications solving specific problems, like steaming out of the doldrums, but failing to grasp the new paradigm, the completely new arrangements that the technology makes possible. Many educational uses of networking are still stuck at steam sailboat technology—perhaps even in the doldrums.

TERC, an educational R&D company that focuses on math, science, and technology in grades K–12, has a long-standing commitment to developing new paradigms of education, including applications of technology to support project-centered learning. This paper summarizes TERC's experiences in implementing one new educational paradigm, the Network Science curriculum model. This curriculum model makes use of the technology of data exchange. The paper addresses some of the technical issues that have arisen in implementing the model. In particular, the relationship between data and their context are examined, and a format is proposed that will enable multiple users to share not only data, but the context of the data as well. Finally, a standard for data exchange is outlined, and we propose steps toward the definition and widespread use of this standard.

2 New Models of Network Use

New models of network use are emerging which make use of the power of the Internet to involve students in computer- and network-based collaborations that provide them access to rich human, informational, and computational resources. The Testbed for Telecollaboration is developing one such model, *Network Science*, characterized by:

- technology-enhanced projects
- shared data
- shared goals
- shared knowledge building
- collaboration
- investigation of real science problems

The model which has emerged from our work with the various Testbed communities is grounded in TERC's work over the past decade in developing the educational power of technology, including the development of the NGS Kids Network® and the Global Laboratory curriculum. With the emphasis on technology, shared data, shared goals, and shared knowledge building, this model also reflects the way that many scientists work.[1]

The Testbed staff at TERC has documented the Network Science model[2], described the results of this model as applied to various Testbed communities[3], and collaborated with the staff of various Testbed communities in writing curriculum that reflect this model[4]. The Testbed staff has also created technologies to support Network Science, including the Alice Network Software client (http://hub.terc.edu/terc/alice/alice.html) and an automated data sharing server (http://hub.terc.edu/terc/alice/newsletter2).

Students' access to data is key to good investigations. As telecommunication becomes less expensive and good connections become widely available in schools, science classrooms can share data which students have collected themselves. By creating sets of data collected at multiple sites, students can evaluate the significance of a larger data set. Data collected from geographically distributed sites allow students to look at patterns that would not be accessible in any other way. Because the data are collected as needed, students are able to refine their hypotheses and take additional measurements. Finally, students are able to engage others (students, teachers, scientists—all online) in interpreting the data—and through these discussions experience first-hand the excitement of science.

While data exchange introduces a variety of new possibilities, it also introduces new problems. Because the data no longer reside with the person or group who collected them, the organization and format of the data, and the context which gives meaning to the data, are in need of explication. Contrast that to locally collected and locally used data, where the organization, format and context are created by the teacher or by agreement of the students, and by common (often tacit) understanding of procedures. This paper addresses the need that has become apparent to us to attach to the data the context which gives meaning to the data.

3 Data and Their Context

Context helps people give meaning to data. Consider a sample data set[1]:

Sue	female	1.73	1.73
Jamal	male	1.70	1.79
Enrico	male	1.60	1.65

Table 1: Data

To make any sense of these data, we need to know what the names and numbers represent. If we add column names, we begin to get the idea.

Student	Gender	Height_1	Height_2
Sue	female	1.73	1.73
Jamal	male	1.70	1.79
Enrico	male	1.60	1.65

Table 2: Data with Column Names

We use the term meta-data to refer to the information that gives context, and therefore meaning, to data. For each column, meta-data includes formatting information such as the data type and length. For each data type, other information may be required. For example,

type=decimals

requires information about precision, such as

precision=2

(number of decimal places); for categories like "gender," information about what categories are permitted is needed:

type=category
options="male","female"

Each column may also include fields intended to give the user additional information. We have found use for an annotation field and a reference field. An annotation field describes the data in a column in more detail than the column title. This annotation could be displayed by the user's software. For example, the column Height_1 might have the annotation, "Student's height in September, measured in meters. Measure to the nearest cm." These words would appear when students had the pointer on any cell in this column. A reference field is a pointer to some source of information about how to measure this category. If this reference were written as a URL, then fully-enabled software would access the reference upon request. This information would be represented by

annotation="Student's height in September,
measured in meters. Measure to the nearest cm."
reference=http://hub.terc.edu/
curriculum/changing_bodies/ height_1.html

In addition to meta-data about a column, meta-data can describe the table itself. For the table, meta-data would include an address for the server location of the consolidated information (master table) and possibly an annotation for the whole data set such as "Data collected by sixth grade students in the state of Massachusetts, starting in 1993," with a reference that points to the curriculum that motivated the data collection activity. As before, if the reference were written as a URL, then fully-enabled software would access the reference upon request.[2]

[1] The data table design used in the example is for column-oriented data, typical of a database, and not for other data structures. The implicit design is that a similar set of data is being collected repeatedly, typically by classes at different locations and/or at different times. In the simplest case, a group of classes is collecting information about a set of variables, such as acidity of the first rain in May, and analyzing their consolidated data set. A more complicated design is that successive groups of classes are collecting a set of data such as the first appearance of a certain animal species in the spring, and the consolidated data set can be analyzed for a single year for geographic variation, or at a single location over a succession of years.

[2] In our design, described below, we have also included the possibility of including meta-information about each row. To keep the discussion simple, we are not

Meta-data

Meta-data is information provided with a data set that gives context to the data, for both the system that processes the data and for the user. Some meta-data can be used by the system (client and server software) to process and validate the data. Some meta-data can be read by the user. Both the table as a whole and individual columns can be assigned meta-data.

For a column, meta-data includes formatting information for the data in the column such as the data type and length, a reference, and an annotation (additional information for the user that is descriptive of the column).

For a table, meta-data can include the URL of the master table for the consolidated data set, a reference (a pointer to where additional information can be found), and a general description of the data set.

4 Evolution of Data Exchange Technologies: TERC's Experiences

TERC has designed technologies to support Network Science projects including NGS Kids Network, Global Laboratory, and multiple projects in the Testbed for Telecollaboration. This experience has shaped our understanding of data exchange and analysis, and is helping us move our design towards a simpler, more flexible, and more versatile data exchange model.

The original data exchange model was developed in 1988 for NGS Kids Network. We refer to this model as a "closed" community because of the many technical requirements for participation. It involves a community with pre-defined data in a specially-designed file format. The data are exchanged among classrooms using e-mail. Each curriculum unit has its own specialized client application, which is used by all classes participating in that unit.

This model is sufficient for the pre-defined, unchanging use of data in NGS Kids Network. It is an approach analogous to the prevalent practice in business of developing a specialized database application for each data base. However, it became clear that this model was not flexible enough to extend to a wider audience with diverse projects and data needs, as required by Network Science curricula.

The model of data exchange we currently use (the Alice model) is more flexible and more open. It was specially designed to be cross-platform (Mac and PC), work on low-end Macs as well as PCs running Windows, and to connect schools via e-mail. In this model there is a general purpose client application, the Alice Network Software, which can be used by all projects for data exchange and data analysis, and a server that provides automated data sharing. The client and server use ASCII e-mail messages as a means for data exchange. The comma-delimited ASCII data format for submitting and retrieving data is intended to open the possibility for participants to use any e-mail program for data exchange, and to use any data analysis tool for analyzing data. The Alice Network Software uses a specially-designed file format, one that allows for annotation, data types, and formatting for each column, and the software can import and export data in both comma- and tab-delimited ASCII and in WKS format.

The Alice model has been used by Global Lab and other Testbed for Telecollaboration projects for the past two school years, and is designed for a low-bandwidth environment. The model has enabled groups of classrooms to consolidate data sets that they have collected, to analyze their data, and to discuss their findings. The software includes some uncommon approaches to data collection (e.g., use of latitude, longitude, categorical data) and to data analysis (e.g., the creation of graph types designed specifically for Network Science projects).

In order for this model to work, several pieces must be in place. First, users (classrooms) must have templates for the Alice Network Software that provide meta-data for each column: annotations, data type, and formatting information. These templates, usually sent with the software to each school, are used both for recording data before submitting it to the data sharing server, and when data are retrieved, for displaying the ASCII data retrieved from the data sharing server. Second, users need to have a special address book for data submission/retrieval containing database address and codes, and to correctly select the name of the database to which they are submitting. Third, the server must have in place a database that matches the data types being submitted by classes.

Through our Testbed projects, these issues have become evident:

- It is easy for users to submit data to the wrong table by making an incorrect selection from the address book; not only easy, but common!
- Data templates are complex, and often project organizers or participants want to make improvements. However, to make changes to a data template after the software is released requires users to have a conceptual understanding of templates; this makes the process of changing a template (or the curriculum) after its release very difficult. Some changes in the template may also require a change in the server data table.
- The current design only allows submission of new rows of data to the consolidated table, and does not allow us to specify when a row of

---

discussing the uses for this information. Such a discussion has to take account of the process of data consolidation on the server.

data being submitted is an addition to previously submitted data, or a correction to previously submitted data. In practice, each of these possibilities is important for different situations.

- While the original system was cleverly designed to allow users to participate fully using any e-mail client (e.g., classes abroad, where they did not have access to EcoNet or other Alice-enabled telecommunications system), it was in fact only a few technically savvy and adventuresome users who attempted to do so, and even for these users, the chance of them making an error was very high. The goal of an open system affording widespread accessibility was not achieved.

In short, the system has proven in practice to lack sufficient flexibility for widespread, scaleable classroom use, to require more training in use than most teachers are likely to receive, and to be prone to human error. As we analyzed the problems, we recognized that understanding the role of meta-data as part of data sets was one key to solving the problem. A redesign of the system had to begin with better distribution of meta-data.

5 Redesigning the Data Exchange System

We are currently designing a new data exchange model whose goal is to facilitate data exchange among users who may be using different software, and to make possible analysis using a variety of different data analysis tools. Key to this approach is a proposal for a new data exchange format.

5.1 Keeping Meta-Data With Data

The basic principle is to keep the meta-data with the data itself at all times. The meta-data includes information about the table as a whole as well as information about the data itself.

In the new system, the meta-data will be stored on the server with the data. When users want to collect data, they query the server for the meta-data and are sent the information from which the client program can automatically configure the software (i.e., create the template). In this way, the template will always be the latest revision, and there will be fewer constraints on revising the templates.

When users are ready to submit their data, the user can simply tell the application to submit the data. There is no need to select an address from an address book; the data table already knows the address of the master table, which is part of the table's meta-data. When the server receives the data, it can read the name of the table's master table and parse the meta-data in order to determine the right actions to take with the data (e.g., whether to add the data as new rows or as a correction to old data). Further, since the data type is also included in the meta-data, any data type errors will show up as mismatched meta-data, which can then generate an error or be accepted as appropriate.

The meta-data is included when a consolidated set of data is retrieved from the server, so there is no need for users to have on hand a template to view the data appropriately.

In fact, the inclusion of the meta-data with the retrieved data makes the system more open and flexible in another very important way. In addition to making data exchange easier to accomplish, this design will also make the use of multiple tools for data analysis much easier. Whereas the current Alice model is designed specifically for use of the Alice Network Software data analysis tool, the presence of the meta-data opens the possibility of users importing the data easily into a data analysis tool of their choice. This choice may reflect what is locally available, or more ideally, will reflect the kind of analysis they want to do with the data. Although the existing Alice Network Software data analysis tool is a very useful general purpose tool, teachers or students may already be familiar with a particular spreadsheet application, or they may want the additional power of a specialized application such as a mapping (GIS) application.

Keeping the meta-data with the data at all times is one of the key design features which distinguishes our new model of data exchange. The availability of meta-data means that the client and server software always know how to process the data, and that users always have additional information about what the data actually means, e.g., how the data is collected.

5.2 Proposed Data Format

A common model for data, one with which the Testbed has worked, is a column-oriented data table. In this model, data is organized into columns and rows. Each row represents the data collected by one class; and each column represents data responding to the same question, such as the height of the student, and therefore has the same data type (decimal number) and the same units of measurement (meter).

A common exchange format for data of this type is tab-delimited data. We propose a new format, which we call **Meta-Data Enhanced/Tab-Delimited data (MDE/TD data)**. This format consists of tab-delimited fields, and fields can include meta-data and markers in addition to data itself. Meta-data and data have been defined; markers are words that are included to help software applications to parse (and people to understand) the fields of information. The markers in the diagram below are the words **meta-table**, **meta-column**, and **data-row** (shown in **bold**). When markers are not provided, defaults will be used which will allow tab-delimited data sets to be handled appropriately.

The marker **meta-table** must appear in the very first field, and indicates that all fields in these rows contain meta-data about the table. The marker **meta-column** shows which rows contain meta-data about the columns. The marker **data-row** tells which rows contain data. In Table 3, we are showing how table meta-data and column meta-data are saved.[3]

meta-table	Table meta-data	Table meta-data	…
meta-column	Col 1 meta-data	Col 2 meta-data	…
meta-column	Col 1 meta-data	Col 2 meta-data	…
meta-column	…	…	…
data-row	data	data	data
data-row	data	data	data
…	data	data	data

Table 3: MDE/TD Format

The primary benefit of this design is that, with no additional work, all standard programs which currently support tab-delimited data can read the file. In the simplest case, the user can read the formatting information and configure the data table. Looking to the near future, new programs and translators designed to take advantage of the meta-data will behave sensibly. Fields that they can interpret will be used to automatically configure the application; and fields that cannot be interpreted by that particular application will appear in cells as text, enabling the user to optimize choices. Looking ahead even further, applications themselves will be able to optimize the configuration for the user.

What does meta-data look like? Let's look at the example that we have been developing.

meta-table	annotation="Student's height in September, measured in meters. Measure to the nearest cm."			
meta-table	reference=http://hub.terc.edu/curriculum/changing_bodies/height_1.html			
meta-column	type= text	type= category	type= decimal	type= decimal
meta-column		options= "male", "female"	length=4	length=4
meta-column		reference =(a)	precision =2	precision =2
meta-column			reference =(b)	reference= (c)
meta-column	name= Student	name= Gender	name= Height_1	name= Height_2
data-row	Sue	female	1.73	1.73
data-row	Jamal	male	1.70	1.79
data-row	Enrico	male	1.60	1.65

Table 4: Data, Meta-Data, and Markers in MDE/TD format (Note: column references omitted in this diagram due to length.)

The first two rows, each beginning with the marker **meta-table**, give information about the table itself. The next four rows, each beginning with the marker **meta-column**, give information about the columns; and the rows beginning with the marker **data-row** contain the data from the original table. To see the actual ASCII version of this file, click on URL http://hub.terc.edu/terc/alice/format.html.

6 Proposed Approach to Data Exchange

The proposed MDE/TD data format is best understood in the context of other data exchange formats. We will look at two other formats, discuss some general considerations about how to design a data exchange format, and conclude with specific steps that can be taken towards creation of a widely accepted data exchange standard.

6.1 Smart Formats and Standard Programs

Previous approaches to solving the problem of data formats for data exchange have focused on two solutions: standard formats and smart programs/translators. Each of these approaches solves part of the existing problem and introduces new problems.

[3] In order to keep the discussion of MDE/TD format short, we are not discussing how to represent meta-information for table rows. However, in the general model, we are including the option of meta-data for rows as well. Such information might include indicating which rows have been retrieved and which ones are being submitted; and for rows being submitted, which rows are new, which rows are corrections to existing rows, and which rows should be deleted. We have also chosen not to complicate the discussion with a description of meta-data for a cell of data, but the general design will include this option as well.

Standard formats, such as Hierarchical Data Format (HDF) developed by National Center for Supercomputing Applications (http://hdf.ncsa.uiuc.edu:8001/HDFIntro.html), have been created to help solve the problems of data exchange. The difficulty with standard formats, however, is that most existing applications cannot read the file format and new applications (or translators) must be developed to interpret the data format. The HDF library makes support of the HDF format easier, but applications still must be created or modified to support HDF. Additionally, none of the current HDF object formats are flexible enough to support all possible data types (e.g., latitude and longitude, and categorical data), nor do they support the range of meta-data used for data exchange, especially meta-data about the table itself.

The smart programs approach also solves some of the problems. The use of smart programs allows all the existing data to still be accessible, existing programs can be used, and the original file retains all of its detail. This solution, however, requires translators to be created and in the case of FREEFORM, developed by the National Geophysical Data Center (http://www.ngdc.noaa.gov/seg/freeform/ff_intro.html), companion text files containing descriptions of the data sets need to be created. Additionally, there is still the problem of the current formats not being flexible enough to support the range of data types and meta-data.

In this paper, we are proposing a different approach, an approach which uses smart formats, can work immediately with many standard programs, and can serve as a standard to be built into software in the future.

6.2 Designing a Data Exchange Format

What makes a good data exchange format? A good format makes it easy to accomplish scientific and educational goals by sharing data. In other words, it makes it easy to share data, it allows for flexible uses of data, and gives users important support to guide their investigations and collaborations.

We have come up with three general principles that are guiding our current work in designing the format. Those three principles are standardization, versatility/extensibility, and simplicity.

Standardization is important to encourage developers to write software that supports the proposed format. The need is apparent: there will always be users who either can't use a given software application, either because they don't own it, have technical constraints, or have other tools they are accustomed to using. However, if a data exchange format emerges that is generally supported by developers, people would be able to continue to use their favorite software and still be able to share data. In a wide-scale collaborative environment, there is a diversity of software, and professional development resources are scarce (and are best spent on pedagogical ends), so standardization of data exchange format is especially valued.

The format for data exchange should also be as **versatile** and **extensible** as possible, allowing it to meet needs whose importance may not be apparent to us yet, but will be as the format is used. A versatile format is much more likely to become standard since it will address more people's needs. Also, a flexible standard could support special features that some tools provide and others don't. For instance, a mapping tool would include latitude and longitude as data types, but a regular spreadsheet tool does not. By allowing the mapping tool to add these new types but using a simple data storage format, we can simultaneously support tools using advanced mapping features and allow users with less sophisticated tools to read the data and do the processing that their tools do support.

Simplicity means that information is not encoded, but is readily decipherable. This quality makes it practical for others to understand and implement the standard, and to extend it when needed. It also makes it possible for users to work with the meta-data even if they don't have tools that support it properly. For example,

```
type=decimal
length=8
precision=2
```

can be understood by users without difficulty, and allows the possibility of users configuring their application manually.

6.3 Creating a Standard

We propose that the MDE/TD data exchange format be developed into a standard that will serve the educational community, and possibly a wider community as well. We see five steps necessary to reach this goal:

First, we are using this paper to make a case for the importance of having a standard; our intended audience is people working with data exchange in the educational community and beyond. While we are primarily concerned with exchange of data in the context of education (e.g., Network Science projects and similar projects run by GREEN and I*EARN), we are also interested in getting input from the wider technical community.

Second, we will post an initial draft of what such a standard might look like, and invite written comments from others. We are interested in what problems in data exchange they have encountered, and how the standard might take account of these problems.

Third, we will continue to develop the proposed standard to take account of problems that we haven't addressed yet, e.g., who has permission to change meta-data in a file.

Fourth, we plan to adapt the Alice Network Software data analysis tool as the first application to make use of this approach. We will try out the approach in a community using both the Alice Network Software and other data analysis tools, and learn from user feedback.

Fifth, we plan to develop a consensus of users and developers around the standard, with the goal of having the standard widely accepted. To develop this consensus, we will start with the educational community but will consult with others as well. In this way, we will encourage widespread adoption of the standard.

6.4 Future Steps

In the long term, we plan to work with other organizations to put the key pieces in place. These organizations might include groups like CNIDR, I*EARN, and BBN, which do technical development for education, and software development companies. The pieces that will need attention include libraries for working with the MDE/TD data format, translation programs to convert MDE/TD data sets to other formats, and adaptation of existing data analysis tools to this standard.

References

[1] The Collaboratory: Pacific Northwest Laboratory's Environmental and Molecular Science Laboratory. (1995, April). *Scientific Computing and Automation,*. pp. 49-51.

[2] Feldman, A., & Nyland, H.. (1994, April) *Collaborative Inquiry in Networked Communities: Lessons From the Alice Testbed.* Paper presented at the AERA, New Orleans, LA. Cambridge, MA: TERC. Available from the authors, or http://hub.terc.edu:70/hub/owner/TERC/centers/tfl/Alice/Testbed.

[3] Feldman, A., & McWilliams, H. (1995). *Planning Guide for Network Science.* Cambridge, MA: TERC. Available from TERC, 2067 Massachusetts Ave., Cambridge, MA, 02140, or http://hub.terc.edu/terc/alice/netplanning.html.

[4] McWilliams, H. (1995). *Wetlands: An Environmental Science Telecommunications Curriculum.* Cambridge, MA. TERC.

McWilliams, H. (in press). *EnergyNet: An Energy Education Telecommunications Curriculum (tentative title).* Cambridge, MA. TERC. Both curricula available from Testbed for Telecollaboration, testbed@terc.edu, or mail: c/o TERC, 2067 Massachusetts Ave., Cambridge, MA, 02140.

Author Information

Alan Feldman is co-Director of the Tools for Learning Center at TERC, and Project Director of the Testbed for Telecollaboration. He is learning to enjoy hanging out with techies, who are a very different group than the teachers and kids with who he used to hang out. Current address: TERC, 2067 Massachusetts Ave., Cambridge, MA 02140.

Lisa Johnson is Technical Director of the Testbed for Telecollaboration at TERC. No average techie, she has a masters in Educational Technology from Harvard GSE and loves to play soccer on Wednesday afternoons. Current address: TERC, 2067 Massachusetts Ave., Cambridge, MA 02140.

Daniel Lieberman was a Unix Systems Manager at TERC, where he worked on the staff of the Testbed for Telecollaboration. He is now headed west to NETCOM, where he will be pioneering the future of the Internet. Current Address: NETCOM, 3031 Tisch Way, San Jose, CA 95128.

Irene Allen is a Software Engineer at TERC, and works on the staff of the Testbed for Telecollaboration. She focuses her efforts on data analysis tools. Current address: TERC, 2067 Massachusetts Ave., Cambridge, MA 02140.

Johan van der Hoeven is Software Architect at TERC, and works on the staff for the Testbed for Telecollaboration. He is also an artist, some of his work can be viewed at:
http://www.rosebud.com/cgi-bin/johan.
Current address: TERC, 2067 Massachusetts Ave., Cambridge, MA 02140.