

TAMS Zine

#1, Summer 05

Notes on the State of TAMS

This has been a busy spring here. After over six months of sitting back and working on other things, now that I have tenure, I have returned to refining TAMS Analyzer to the best of my rather sorry coding ability. This has meant adapting TA to a client/server model (while not forgetting single users), creating powerful new reports, expanding Graphviz integration, international character support, and generally fixing bugs.

What remains? First of all there are a number of details that still need work in the client/server system including some sort of memoing/email system. Also the ability of administrators to remove files from a project needs to be

worked on. At the moment, files can be added, but never removed! There are also a few bugs floating around that I have to acknowledge, especially in the regex system, which probably needs a good updating anyhow.

TA has successfully made the transition to Apple's 10.4, though it has meant some problems for Jaguar (10.2.8) users. These hopefully have been fixed at this point. Most of these problems have come with MySQL support. The framework seems to have had an odd bit (setuid) set which has caused some problems for early system users.

What of the future? Well Apple is moving to Intel, and I will

Continued on page 4

sporatically released news, stories, announcements, tips, and prognostications regarding tams and tams analyzer.

contents

- **state of tams, p. 1**
- **struct and unstruct, p. 1**
- **data comparison table, p. 2**
- **open source tools, p. 2**
- **please contribute, p. 3**
- **consulting, p. 4**

Features: Using Struct and Unstruct

Print qualitative data has multiple different forms: field notes, interviews, documents, open ended surveys. TAMS supports two ways to handle documents to add flexibility for different types of documents: structured and unstructured.

Initially TAMS only used structured documents, so I'm going to begin there. Structured documents can be conceptualized as a series of filing cards with a standards set of context variables filled in afresh on each. TAMS's

first assignment was coding newspaper articles. Each one was independent of the one before it. Each one contained a variety of identical context information at the start (author, title, date, etc.), and each terminated the same way. So I used an `!end` tag, inserted through search and replace, and the context values were cleared when each `!end` was encountered. Other structured data include emails and surveys. Inter-rater reliability should be done with a structured document. If you want a

document to be treated in a structured manner use `!struct` at the top of the document.

The problem is that this does not work with all types of data. Interviews, in particular are better thought of as a flow of context and non-contextual data. Context values (who is talking, the current time code) keep changing as you move through and it is TAMS's job to just attach the current state of the `!context` values to whatever stretch of coded data has just terminated. This is

Continued on page 3

Tips: Merging Columns in a Data Comp. Table

Data comparison tables offer a powerful way to display qualitative and quantitative information. They are created by prompting a researcher for a category that serves as columns, a category for rows, and, finally, for the content of the column/rows intersections.

One problem with data comparison tables (DCT) is that if you chose "Codes" as the content of your columns, the data or counts you wish to examine are scattered across different columns. Often, after I have carefully

"speaker" compared against "Codes"

speaker	code1	code2
One	quote 1	
Two		quote 2

selected data I simply want to view the data (or counts or some other data) in a single column. At first blush there is no way to do this: I have to commit to codes, code sets, or another results window column for my DCT columns and these

inevitably have varied data in them thus results are spread across multiple columns.

By taking one additional step however I can get the results I am seeking. All I need to do is to create a column in my results window that contains the same

value for all records. This needs to be done before evoking the DCT option from the Reports menu.

Step 1: Add a column

With my search results as the front window I pick "Results->Temporary column->Add temporary column...". From the sheet that pops up I'll need to assign the column a name and a string to fill in the table for both the default and selected values. The string should be what I want this single data column titled. Click "Ok". I typically use the phrase "All" for the three values: the column name, the default value, and the selected value.

Step 2: Start the DCT

Once I have a table column filled with the value that I want as the

Continued on page 4...

The Best Things in Life are (Kinda) Free

OS X is blessed to sport all manners of free software. I want to spend a few column inches sharing with TAMS users some of the gems I have found invaluable in doing my work as a teacher and a researcher. The programs I am listing here are all "Aqua" programs, i.e., they are native Macintosh applications. There is a whole parallel world of free software available for X11, the unix windows system available through Fink and other port maintainers. Links will be provided at the article's end.

The first two programs I am going to discuss are especially important to laptop users. TAMS is developed and maintained on my tiny PowerBook G4, provided by my university, so I'm sensitive to the limitations of a laptop. First there is very limited screen space,

and I like to use multiple programs at once. Typically when documenting TAMS I will have open Word, TAMS, Canvas, and a browser (and maybe my mail program!). To keep things straight it's easiest if each application is on its own "screen". Desktop Manager to the rescue! Basically, each of those applications gets its own desk top. Each acts as if it were the front window, there's no looking through windows! I just click the desktop I need from a little floating palette and that program leaps to the front and all other programs vanish. DM has wonderful effects for shifting screens as well.

The other thing I use constantly is iScroll2. This adds a scroll wheel function to my little track pad. By using 2 fingers on the pad rather than just one, I can

scroll whatever window is under the mouse. Newer PowerBooks come with this feature built in, but for those of us with older equipment, this program provides the same functionality. It has taken a while to get all the bugs out of the iScroll2 system, but recent releases seem steady and reliable!

One of the pieces of free software I am most excited by is the new aqua version of Scribus. Scribus is basically a Pagemaker style layout program, available to Unix/Linux users for a couple of years. Finally, in the last two months, initial ports of the program for Macintosh have been made available. By the way, this newsletter was completely done in Scribus-aqua-version. I have other layout programs but I have found them to be user-interface

Continued on page 3...

TAMSZine: Issue 2 Depends on You!

TAMSZine is an experiment in providing support and community to OSX and GNUstep users in an informal, collective manner, i.e., in the spirit of TAMS itself. For there to be a second issue, you will need to contribute. I want stories. Tell the world about how you are using TAMS to solve problems. I get to hear from users about all sorts of wonderful projects: there are a lot involving medicine, several involving race, one person I know is doing the anthropology of national security (aside from myself). Let the world know what you are up to.

I also want hints. How have you solved a particular analytic problem using TAMS? What features do you find the most useful? Are there ways you've used features in unconventional ways that others may find useful?

This is a zine, which is defined as an unconventional, informal magazine. Grammar is optional. Spelling helps, but I'm not a stickler. Your work is very important, however. So please share it! Send contributions to mweinste@kent.edu.

Struct & unstruct continued...

unstructured data. To force data to be considered unstructured put a `{!unstruct}` tag at the top of the document, which is now the default behavior.

The problem with unstructured data is that it has no clear boundaries. As a result certain types of information cannot be pulled out of unstructured documents, mostly of the form "show turns of the conversation that deal with code X." I can find all the code X's out there, but I can't find turns of the conversation because a turn is a structure and the data is unstructured.

So there are ways of treating interview data as structured, and some tools to automate the process. In my work I find I have typically use two context values in interviews: a time code and a speaker code. I

automate the process by indicating to TAMS that when it sees a speaker code, it should be treated as the last contextual information it will see for this "turn". Any new context code it encounters should be treated as if it had an `!endsection` tag before it. To specify this I place at the top of the document the tag `{!last speaker}?note !last` has to be used on a document by document basis; you cannot put it in your init file and expect it to carry through to the rest of your documents.

Using `!last` has consequences for how I enter my transcript. For example, any time code must be set prior to the speaker code for that turn. It also introduces its own problems. Consider an interview where the person talks for five minutes uninterrupted. In theory this is one

turn, but you would want to give it many time codes. Unfortunately, the `!last` compromise will not work in this case. You will have to artificially break this into a number of turns if you want to assign intermediate time codes. The alternative would be to "manually" insert `!end` tags to mark turns in the transcript and use the `!endsection` tag to mark the intermediate points where you put time codes.

By using `!last` each change of speaker will be treated as if on a separate card. Now I can ask for things like "show me all turns that talk about X and Y" (search for "X+Y" using a section search) in this interview. X and Y do not need to overlap here, as long as they occur in the same stretch of text described by a value of "speaker".

Free software continued...

disasters. Adobe's Indesign rather stands out in this regard. Scribus is simple, fairly (not 100%) intuitive and quite powerful. Readers should be forewarned; it is still in a very experimental phase. You have to drop a whole lot of stuff into your `/Library/Frameworks` folder for it to work. Hopefully soon it will be available in one binary package. Until then it is perhaps only for the digitally brave. That

said, I am delighted that it's available!

Finally, a quick note about html editors. The best of the category is under X11, called Bluefish. There are a couple of Aqua programs out there. For instant gratification of simple pages that I don't care that much about the quality of the html I use Nvu, which just reached version 1.0. It's a "wysiwyg" program. For

lower level page design I have been using Taco HTML. It's very simple, and designed for people who want to work at the tag level, but it's also very solid.

Links:

<http://desktopmanager.berlios.de/>
<http://www-users.kawo2.rwth-aachen.de/~razzfazz/>
<http://aqua.scribus.net/>
<http://www.nvu.com/>
<http://www.tacosw.com/>

Data comparison tables continued...

title of my single data column, I pick "Reports->Data comparison table".

Step 3: Set the column, row, and data

From the "Compare" box, I pick "Other column". A new pop up menu will appear which lists the columns of my results table. I pick the new column I created (in my example "All"). From the

"speaker" compared against "All"

speaker	All
One	quote 1
Two	quote 2

"With field" box, I pick the category I want to use to break up the column vertically, i.e., what do I want the rows to represent (each document? each speaker?). Finally, I pick the data I want to examine by selecting it from the "Data elements" box. Remember you can select multiple items and rearrange the items by dragging them into the order you want them listed in the table.

State continued...

be making fat binaries that work on the new systems natively. At present I'm waiting for "someone" (MySQL or someone else) to compile a fat version of the MySQL client. At that point it will be easy, since TA is mostly straight cocoa. It's the add-ons that need to be converted: AGRegex/PCRE and SMySQL. Once those have been converted, recompiling TA is just a matter of 10 minutes of compile time!

Support Open Source: Let Me Train Your Crew

You use TAMS, you like TAMS, you have even offered to wash my car and mow my lawn each week (see the TAMS page on macupdate.com). Now you can support TAMS and get a real bang for your buck. If you are at an institution that uses TAMS bring me in to do workshops at the beginning, intermediate, and advanced levels. Price is negotiable, but I am at the point that TAMS is going to have to earn a little bread in our declining income situation (health care is going

to zap Kent Faculty in the new contract, whatever millenium it's approved).

I'm willing to travel (living in Ohio I might add that I love to travel) and I can add on any number of fascinating talks about science education (via critical theory) and the anthropology of science (especially, the public culture of human subjects, but also science museums). Drop a note at mweinste@kent.edu!

TAMS and TAMS Analyzer are distributed through a GPL license. The binaries and source code are available at <http://tamsys.sourceforge.net>

TAMS Analyzer includes:
AGRegex © Aram Green
PCRE © Philip Hazel
SMySQL Framework © Serge Cohen

Issue one of TAMZine is written by Matthew Weinstein
© 2005, Matthew Weinstein, All rights reserved, but feel free to share this with others

Contact Information
Matthew Weinstein
404 J White Hall
Kent State University
(330) 836-4452
mweinste@kent.edu