

# Project Juggling – The Evolution of a Process

Joy Hathaway  
Fermi National Accelerator Laboratory  
P.O. Box 500  
ms 367  
630-840-3649  
Hathaway@fnal.gov

## ABSTRACT

What do you do when an enterprise project management system is too much and a list of projects is too little? Our solution was to create a simple Access database with a web interface using ColdFusion. Our objective was to facilitate the CIO's (and other managers') coordination and understanding of the full range of projects (about 75-100 active), organize periodic status reports for all projects, and automate the flow of detailed effort data which was required by the CEO. (The status reports, themselves, are stored in a free document database system.) Our "quick prototype" turned into a useable system for tracking -- on a "medium grade" scale -- what was being done, by whom, when, and why. And, in the end, answering the big question:

How much effort is this costing us?

## Categories and Subject Descriptors

Computer Applications J.1 [Administrative Data Processing]: Government –

## General Terms

Management

## Keywords

Project management, Dynamic web application, ColdFusion.

## 1. INTRODUCTION

Fermi National Accelerator Laboratory is a government high energy physics research facility operated by the DOE under the direction of the Universities Research Association -- a consortium of 90 universities with physics research programs. Fermilab builds and operates the accelerators, detectors and other facilities that physicists need to carry out forefront research in high-energy physics. Fermilab is the largest high-energy physics laboratory in the United States. There are approximately 2200 people working at Fermilab, including Cooperative students. Fermilab also has almost 2300 users. A user is a researcher from an outside institution who uses Fermilab to carry out research in particle physics or related disciplines.

Copyright 2003 Association for Computing Machinery. ACM acknowledges that this contribution was authored or co-authored by a contractor or affiliate of the U.S. Government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.

*SIGUCCS'04*, October 10–13, 2004, Baltimore, Maryland, USA.  
Copyright 2004 ACM 1-58113-869-5/04/0010...\$5.00.

Our research is very data intensive. In March the Computing Division achieved a new milestone of accumulating one petabyte of data. On an average day at Fermilab, approximately ten terabytes of data are transferred from experiments to robotic storage. Our networking requirements are also extensive. In a large part, this is due to data transfer but it is also crucial for the communication between collaborators who are spread across many countries.

Our Computing Division has approximately 280 employees. The size of our organization makes the task of coordinating all the projects and associated effort an enormous undertaking.

## 2. MOTIVATION

We currently have approximately 75-100 substantial projects in the Computing Division. For each project, the group (or team leader submits) a "project definition" which includes the name of the project, category of the project, project leader, stakeholders, deliverables, estimated start and end dates, description, and any associated URLs. Information that changes frequently is supplied via URL to make it easier for the project leaders to keep them current. URLs are given for: project website, documentation for deliverables, project plan, and outstanding issues.

The project leaders give periodic (ideally, once a quarter) status reports. The purpose of these reports is to update interested parties on the progress, issues, and plans for the project. These status meetings have a reserved weekly slot on the Division calendar. They are attended by the department heads, deputy heads, stakeholders, and anyone working on related projects. The presentations (.ppt or .pdf) are stored in a central location (linked to the database via URL) where they are available as a resource for the Division management and staff.

Our original collection of project information was a group of static web pages -- a list of projects with links to definitions and reports. As we accumulated more data, these pages became burdensome to maintain and difficult to search.

## 3. CHOOSING TOOLS

Our initial decision was to create a quick proto-type (famous last words!) database accessible via dynamic web pages using tools which were currently available and "light weight." The understanding was that we would eventually choose more robust tools for a future version after we had a more clear understanding of the issues and needs. Because of their concerns about security, our systems group was not willing to run a php server on our central systems. Our group had our own IIS server running ColdFusion, so we chose to use Access and Coldfusion.

During the initial design phase, we also began to look into project management software to see if it could be used for all, or part of, the final solution. Enterprise project management software such as Microsoft Enterprise Project Management (EPM) and Welcom offered very powerful tools for balancing project and effort resources. However, this software required each project to supply accurate, detailed information at a level that was unrealistic for our environment. The overhead for each project to maintain the necessary details would have added a crushing burden for our project leaders (who are also active participants in the project).

At the same time we were also forced to choose a new web-authoring tool. HoTMetaL, our previous tool, is no longer supported and couldn't cope with ColdFusion tags. The switch to Dreamweaver was on our "to do" lists but none of us had been able and/or willing to set aside the time to learn a new tool. This project forced the issue because Dreamweaver and ColdFusion are both Macromedia products and are, therefore, totally compatible with one another.

## 4. PHASE I

The first task was to collect the existing data and put it into appropriate database fields. We used the "PMBOK Guide" ("A Guide to the Project Management Body of Knowledge") to keep our terminology consistent with current project management usage. The status reports and closeout information were kept in their current location for this phase; only their URLs were added to the database.

This setup gave us the basic ability to search, sort, and display the project information in more useful ways with relatively little overhead. Next we turned our attention to the data itself. We created web forms for project leaders to enter/update their project definitions and we checked data for accuracy and completeness. We found that project leaders had difficulty filling out the forms in spite of help files. Checking the data was a never-ending job. This would have been a good time to have a general meeting of project leaders and department heads to discuss what the purpose of the database was and where we were trying to go.

### 4.1 Minor Modifications

Originally our database included only projects. By definition, projects have a beginning and an end. There are, however, many computing tasks which never end. We wanted to include these thankless tasks in our reporting process. This broadened the scope of our application. "Project" became a misnomer but we continued to use the word and the same format for data with a somewhat looser definition.

Another feature of our database/web system was an online, database-driven meeting schedule. The meeting schedule with links to the project definitions and reports made it easy for people to decide which meetings they would like (or needed to) attend. We had another type of meetings called briefings. A briefing is a report and discussion about a topic which may or may not be associated with a project. Since it is inconvenient to consult multiple sources for similar information, briefings were added to the status report schedule. Since they had no project definitions these meetings needed titles and abstracts – two additional fields for our database.

Next we wanted to link the project definition with the task number (budget code) so that we could correlate the budget with the projects. Conceptually, this was easy; getting accurate data -- or

sometimes, any data at all -- was extremely difficult. This was probably due, in part, to the general tendency to avoid bureaucracy but was greatly influenced by some amount of confusion about the laboratory's recently restructured accounting system (we are in our first fiscal year of it). In addition, the project leaders were not always the same person as the one who set up the budget structure. We ended up correlating the numbers with the projects using the combined effort of the Head of Projects & Outreach, the department heads, and the Division's financial office.

## 4.2 Documents Database

We were beginning to collect a rather large body of documents. We had the presentations stored in a file structure on our server but they were not easily accessible except through the project definition. We needed a database system for storing these documents. We didn't want to totally segregate the status reports and briefings from the other Division documents so we were looking for something that would be appropriate for any Division documents.

Staff working in some of the experiments at the laboratory had adopted a document management system, DocDB, which had been developed in house (<http://www-btev.fnal.gov/cgi-bin/public/DocDB/DocumentDatabase>). Many of the people in Computing had used it in connection with the experiments and found it to be very useful. We set up our own DocDB server and configured it for our documents. We have had a lot of interaction with the developer as we find its limitations and bugs. We are now in the process of entering documents and evaluating whether or not this will be the best solution for storing our documents.

One advantage of a document management system is that it is easy for people to enter their own documents into the system. This is more efficient and safer than having speakers send us their documents as attachments.

## 5. INCLUDING EFFORT REPORTING

Our previous effort reporting system was a Java-based program originally written for use by one department and, when we decided to have online effort reporting for the entire Division, it was modified to meet those needs. It collected data from each employee stating how they had divided their time (effort) for the current month. The results of this program were put into reports (a separate one for each group) that were then compiled by the Computing Division financials office for reporting to the Fermilab Accounting Office. Currently, only a small portion of the data collected needs to be passed on to the laboratory. However, we expect this to change in the next fiscal year. In preparation for that change, we needed to make sure that people were reporting their effort to the correct activity. Retroactive changes to the laboratory's accounting system are possible (and not encouraged) for only one month.

Correlating the projects with the task numbers will allow us to link budget information with the descriptions and reports. The budget information was already being put into Crystal Reports (via Access tables) for our managers. Having a detailed breakdown of effort in Fermilab's accounting system will assist our managers in tracking how much effort each project is really costing us.

### 5.1 Accounting

We were able to access the Fermilab accounting structure but in order to mesh with the laboratory's accounting system, we had to

adapt our terminology to be compatible with theirs. From their perspective, most of our Division's work is considered one project. Sub-divisions of projects are called tasks. We changed our terminology to call Computing Division projects "Activities" which was also more accurate for ongoing "projects" and also enabled us to group accounting tasks under one "activity." Roughly speaking, what we considered categories were called "Roll Ups" in accounting. Names which project leaders used for talking about their projects were not necessarily recognizable from the task descriptions. In short, there were a lot of details to tend to before the two systems could blend, seamlessly, together.

The department heads set up their accounting so that more than one project might be reporting to a single task number. To give each project its own unique identifier (one that would have a recognizable and sensible relationship to the accounting structure), we went head-to-head with our financials office and declared that we would be keeping the master list, adding to it where necessary. We would provide them with a subset of the list which they would then send to the laboratory's accounting office. This was a sharp departure from the previous flow in which the laboratory kept the master list and our financials office submitted changes to them. From the accounting office's perspective, nothing has changed but it has been a change in mindset for our division.

## 5.2 Personnel Data

In order to validate employee information we had to obtain access to the personnel data. Conveniently, that data was already being maintained by the Computing Division, in Oracle. We were able to link to that data for validation of IDs, usernames, group affiliation, and status. The theory that "the more you use it, the better it gets" held true in the first month when we found errors in the personnel database while trying to reconcile the effort reporting output.

## 5.3 Formulas and Rules

Because we get funding from many resources (most of them within the Fermilab structure) the financials office had to split the effort on one project across more than one task number (budget code). That means that if PersonA worked 50% of their time on TaskZ, 35% of that 50% might be charged to TaskNumber6 and 75% of that 50% to TaskNumber9. In other cases, if PersonG worked on TaskY, their time would be charged to TaskNumber3 – but, if PersonE worked on that same task, their time would be charged to TaskNumber7. In order to charge effort correctly, we needed to codify these rules and formulas.

## 5.4 Integrating with the Existing System

For the first month, after we made the data changes, we used the same interface for effort reporting. We used our database to generate the list of "activities" which employees could choose from to report their effort (the financials office had been maintaining it by hand). The output was recorded in two places: once in the old way as a printable form, and also, as a delimited file we could load into our database. The financials office used the old forms to double-check the generated data.

## 5.5 Post Mortem after First Month

We had only fairly minor problems concerning the input the first month. Users were somewhat confused by the change in the names

of the activities. It would have helped if the managers had been more involved with the data changes and helped their people identify the correct tasks for recording their effort.

We delayed putting the formulas and rules in place. Instead we continued using the old method of having each participant distribute his/her own effort.

Because the output did not have the same fields for each entry, we had some manual effort to combine the entries into a cohesive table. In spite of this we were able to cut the time for preparing the accounting report from days to 2 hours – on the very first month.

We now have all of the effort data stored in a database where it can be viewed, searched, and combined with other data for reports.

The format of the output highlighted the importance of having full control of the data. The Java scripts were not very complicated so we decided to recreate them in ColdFusion. In order to avoid a revolt when we changed the system twice in as many months, we tried to make the new forms as much like the old ones as possible.

## 6. PLANS FOR THE FUTURE

Now that we have made the first steps on our journey, we need to refine our trajectory to get where we want to be in future years. Having made our "rough draft," we are in a much better position to plan for our end product.

### 6.1 Security

Since our "Projects" database now contains more sensitive data, security is crucial. We will be consulting (will have consulted by the time you read this) with our security group and financials office about what level of security is necessary.

### 6.2 Database

In conjunction with an increased need for security, we are looking toward a more secure database application. Since most of our Division data is in Oracle, we will probably move to Oracle at some point. This will require learning new applications and more stringent planning. Fortunately, we have the luxury of a database group with experience in Oracle who can help guide our way (but, unfortunately, don't have a lot of spare time).

### 6.3 Web Interface

Although ColdFusion will interface with Oracle, we now have the option of using php as well. Since they now have php running on our central cluster, we need to evaluate if it will be a positive move for our application.

## 7. ACKNOWLEDGMENTS

Obviously this project required cooperation and input from all parts of our organization. One of the most difficult parts of the project was getting input from all the right people. Although 280 people sounds like a lot to those of you who are at a small institution, we are operating in a very "lean" environment. It has been very difficult to get time commitments from the busy people whose input is so necessary for the success of the project.

We owe a great debt to all the people who patiently (usually) explained and re-explained their business processes until we could understand them and relate them to the project.