2011

Can Transportation Researchers Reuse Project Datasets and Programs?

Tyler Hayes
Portland State University

Lois Delcambre
Portland State University

Leonard Shapiro
Portland State University

Can Transportation Researchers Reuse Project Datasets and Programs?

Tyler Hayes, Lois Delcambre, Leonard Shapiro
Maseeh College of Engineering & Computer Science
Portland State University
{tgh, lmd, len}@cs.pdx.edu

Abstract

Data users involved in research and analysis typically invest a lot of effort cleaning and manipulating their data as they work. Based on this observation, we have investigated two hypotheses: 1) reuse of datasets and procedures is difficult, and 2) the inability to reuse datasets and procedures is primarily due to a lack of documentation. To test these hypotheses we conducted structured interviews with data users asking questions regarding the struggles in their work pertaining to data, their documentation habits, and the importance of documentation. The interviews revealed that the data users rarely reused data or procedures, frequently encountered poor documentation, did not adequately document their own work, and unanimously agreed that documentation was important and that the absence of documentation was the primary cause for a lack of reuse. The results of these interviews have led us to develop a concept of a tool to help data users document and manage their work.

1 Intro

Data is being collected and harvested by thousands of different corporations, organizations, companies, research groups, enterprises, and industries. Each dataset potentially has a different structure, schema, and/or encoding—even for datasets pertaining to the same type of data. For example, one database of traffic incidents might contain location information based on GPS coordinates, whereas another traffic incident database stores its location data as a combination of roadway and mile marker.

With all of this new heterogeneous data being produced and stored, researchers are embracing the resulting opportunities to study and analyze many new phenomena, discover and/or support new theories, solve new problems confronting their field, and otherwise advance innovative ideas using the information that comes from data. However, there are barriers to accomplishing these goals. We hypothesize that one of the most critical of these barriers is a lack of project documentation.

It has been observed in past collaborative research projects with researchers in fields outside of computer science that projects involving extensive use of data require a great deal of researchers' time and energy to "work with" the data to clean and transform it. Further, a lot of the artifacts of research projects such as procedures, semantics, assumptions, and decisions made during that work are typically not documented. According to the majority of the subjects interviewed for this research project, unless there is sufficient documentation by a project's predecessor, there is no way to know what was done to the data involved in the project. The data users on the receiving end can then either reject the data and look for an appropriate dataset from another source (if possible), start over from scratch with raw data (if available), or try to acquire the necessary documentation from the predecessor through e-mails and phone calls; all of these options cause more work for the data user. Further, the data user may end up doing as much work as the predecessor to produce the same results, thus making no use of the valuable time and energy tendered by the predecessor. Therefore, many data users in various venues of research are potentially causing extra work for themselves as well as any successors to their work because useful documentation is not being produced. Absent documentation also significantly reduces the possibilities of doing an accurate comparison of multiple projects within the same area of research.

The goal for this work was to discover what data users did when working with data, what problems they faced during their work, how well they documented their work, and how valuable or important documentation is to them. In order to find this out, we conducted structured interviews with ten data users. The areas of research that

these data users cover is listed in Table 2.

Our method of gathering information regarding the work of data users is described in Section 2. Section 3 details the relevant information gathered from the interviews. Future work is examined in Section 4 based on the results of the interviews, and we conclude in Section 5.

2 Related Work

Data reuse is touched upon in the work by Piwowar [**?**] and Zimmerman [**?**], but both projects focus more on data *sharing*, which pertains to the act of releasing researchers' datasets to the public rather than the difficulties a researcher may encounter when attempting to reuse the released datasets. Zimmerman does however briefly discuss metadata (data describing other data) as documentation, stating that metadata is said to be the key to data retrieval and reuse [**?**].

The importance of documentation has been addressed, not only in data research, but also in software engineering. de Souza, et al. [**?**] explore the continuing trend of the absence of documentation within software engineering. They report that documentation is considered one of the oldest recommended practices for development and maintenance of software. This is related to our hypothesis that there is a correlation between reuse and documentation, but their study is limited to the software engineering field. We broaden this view to cover a multitude of other fields.

More directly related to our work is the research done by Vardigan, et al. [**?**] where documentation and curation of datasets are specifically addressed. They describe the Data Documentation Initiative which is an emerging metadata standard for the social sciences. It is claimed in their work that good documentation is essential to the reuse of datasets by a secondary user. Alice Robbin [**?**], and Gandara, et al. [**?**] extend the idea of good documentation to include processes, and workflow documentation, respectively. The work by Gandara, et al.—published after our interviews were conducted—included developing a prototype system that creates knowledge-annotated scientific workflows. This is related to the conceptual tool we describe in Section 5.

Recent work done by Wynholds, et al. [**?**] is very similar to our work in that interviews were conducted with astronomers who work with an extensive amount of data about their data practices. It was found in their work that a lack of documentation was a significant disincentive to reusing data [**?**]. In addition, astronomers noted that a steep learning curve for integrating sources presented both a challenge and a liability, and that the trustworthiness of data sources was an issue [**?**]. However, that work is also narrow in scope—only pertaining to the work of astronomers. These issues could very well be a common theme among data users in any field of research.

A panel organized by Sam Uselton also divulges data analysts' personal experience in what they consider their work's biggest challenge [**?**]. However, the panel was very specific to visualization, and may not adequately reflect a general view of the work involving data.

3 Method

Among the many venues with which to gather the desired information (in-person interviews, written surveys, e-mail correspondence, phone conversations) we chose to conduct in-person interviews, because it provided an opportunity to see firsthand how the subject works and what sort of barriers they encounter. We also wanted to be able to experience their mannerisms, facial expressions, and tone of voice for things that they were passionate about and we wanted to be able to easily record the audio of the interviews in order to replay their responses at a later time.

After several iterations, we settled on a set of questions and partitioned them into categories: questions about the experience of the subject, specific projects of the subject, revision control, redo/reuse of projects, and closing questions. The second category of questions—those regarding specific projects that a subject has conducted—were the core of the interview questions. They included questions regarding workflows of projects (high level descriptions of the process pipeline involved in a project from start to finish), as well as lower-level details about what was done for each step in the workflow of a project. For the full list of the interview questions, see Appendix A.

We wanted subjects who work extensively with data for the purpose of research or analysis, and who perform one or more of the following tasks: data cleaning, data integration, and data transformation. E-mail requests to participate in our interviews were sent to our non-computer science collaborators and colleagues who not only would be good candidates themselves, but might know others who would fit our criteria. Through this networking we were able to schedule and conduct 10 in-person interviews.

The interviews were executed with an interviewer and an accompanying observer who took notes on a laptop. A small, digital audio recorder was used to capture the audio of the interview; the observer's laptop also recorded the audio as a backup.

The interview subjects were first given several documents. In addition to the formal cover letter describing the research and corresponding interview as well as their rights as human subjects, the subjects received a glossary of terms that were to be used during the interview (such as "data transformation" and "data integration"), a workflow diagram to visually explain the concept of *workflow*, and a document giving several examples of steps that might be taken in a project. These were given to the subjects in order to alleviate ambiguities with our definitions and terminology.

4 Results

In some cases an interview question produced a wide variety of responses with little in common, whereas others resulted in answers that were quite consistent. This section reports who these interview subjects were, as well as the trends, themes, and otherwise interesting information that permeated throughout the interviews of this research.

4.1 Our subjects

Our subjects had a range of educational backgrounds, and came from a variety of professional and academic disciplines, although the majority were involved in the transportation field. This group of individuals constituted a mixture of grad students, professors, and industry professionals. The amount of experience in working with data for research and/or analysis totaled 136 years amongst these subjects, and ranged between 4 years and 20 years, with an average of 13.6 years. Tables 1 and 2 list the interview subjects' educational backgrounds and areas of research, respectively.

Table 1. Interview subjects post-secondary degrees

Degree	Count
BS Civil Engineering	3
MS Civil Engineering	2
PhD Civil Engineering	2
MS Urban Planning	2
PhD Geography	2
PhD Agriculture & Resource Economics	1
PhD Economics	1
PhD Urban Studies	1
Master of Regional Planning	1
MA Geography	1
MA Regional Science	1
MS Transportation Engineering	1
BS Journalism	1

Table 2. Interview subjects areas of research

Area	Count
Transportation	7
Demography	1
Education	1
Health care	1
Social services	1
Water	1

4.2 Workflows and procedures

The term "workflow" in our context means a high-level description of what was done during the process of a research project in terms of procedures (which we called "steps"). However, we were only interested in those tasks which involved working with data. For example, the following sequence of steps qualifies as a workflow: 1. Collect data, 2. Clean data, 3. Integrate, 4. Analyze.

A total of 20 research projects were described by the 10 subjects we interviewed. Out of those 20 projects, only 3 (15%) contained some sort of data collection—the rest, 85%, either already had the data in-house, or retrieved data from existing sources that may or may not have collected the data themselves. The latter case was much more common. Although the subjects did not seem to mind working with the varying formats of datasets within a particular project, some alluded to the problem of sources changing their formats over time, and thus creating more work for the researcher. One subject would have liked to document the workflow and procedures more in one particular project, but "it seems pointless now because of how many different formats the data comes in." Another subject used scripts to automate the work in a project, but those scripts rely on the data coming in to be in a certain format: "As long as the data keeps coming in at that same format, those scripts can be re-run at any time." On the other hand, one script was not robust enough to handle the changes in the state's inventory of transportation data, so some manual work had to be done as a result.

Not only do formats of datasets change out from under the data user, but the semantics can change as well without notice or documentation, much to the frustration of the user. One subject described a project where this problem happened often. This project, which concerned a program that helped welfare recipients obtain a job, consisted of data received by an agency that found itself having to find a different way to collect data, and thus changing its semantics. "The problem comes from changes in policy [by the agency providing the data] so that they are coding the data differently." Regular meetings were held with mul-

tiple agencies to possibly standardize the semantics, or simply not change the semantics at all, but neither approach was feasible. The subject simply stated that providing documentation on the semantics with the data would provide the most benefit.

The majority of the project workflows contained data cleaning and integration. 75% required data cleaning procedures, and 65% contained procedures to integrate datasets. Only one subject mentioned a project where data transformation was explicitly done, while two other projects included the possibility of transformation.

We asked our subjects about which tools they used while performing their projects, and if they deemed those tools to be sufficient. There was a range of tools used by these subjects as shown in Table 3. The number of unique tools (that were mentioned only once) far outnumbered the most-used tool, Microsoft Excel. Although Excel was the dominant tool used among these subjects, its use was not universal; the use of Excel ranged from being the tool that was used exclusively to only being used for data summary visualization. Although Excel is used fairly commonly in some manner, each research project and analysis problem that uses data requires the use of a specific tool chosen by the individual doing the work; there was no evidence of a single tool that permeates throughout all projects and by all individuals. Therefore, we can conclude that refining a single tool (or even a small subset of tools) would likely not make a very big impact in the community of data users.

Table 3. Tools used in projects

Tool	Count
Microsoft Excel	10
SAS	4
Microsoft Word	4
Python (programming language)	3
ArcGIS	3
Microsoft SQL Server	3
Microsoft Access	2
Oracle	2
SQL scripts	2
Visual Basic (programming language)	2
CenEst	1
DOS batch files	1
EViews	1
Foxpro (programming language)	1
Google Street-view	1
Limdep	1
local programs	1
Matlab	1
MySQL (Database Management System)	1
PostgreSQL (Database Management System)	1
Provider-1	1
SCADA system	1
Microsoft Sharepoint	1
Shazaam	1
Solver	1
SPSS	1
UltraEdit	1
UrbanSim	1

When asked whether the tools used by the subjects were providing adequate support for what they were doing, the answer was unanimously "yes". However, two subjects suggested that their methods may not be the most efficient but are the ones that they are most familiar with. Therefore, even though the subjects were happy with the tools they use, it does not imply that there does not exist (or could not exist) better tools and practices. In regard to using the arguably obsolete programming language Foxpro, the subject stated, "It might not be the slickest way of doing things, but it works perfect[ly] [for me]." Another subject mentioned the desire to integrate data into a statistical package as soon as the data comes in, which can be done with the programming language R, but didn't use it because of the overhead of learning R.

Interestingly, there were two subjects that wrote SQL scripts with relational database management systems (PostgreSQL and MySQL) extensively for their projects. One of those subjects explained the reasoning for using these tools: the code of the scripts can be read in order to know what people did including which records were being selected, processed, and cleaned, and the processes are repeatable by simply executing

the scripts. "If you want a small, specific level of automation, Excel macros can provide that, but for bigger automations you'll end up not gaining as much as if you moved out to [scripts]... It's harder for engineering students to break out of the spreadsheet mentality."

The interview subjects spent the most time on the question that prompted them to discuss the details of each project's procedures. There were no two procedures in the projects that were close to identical. Every study discussed was very specific and unique in that they were attempting to solve a specific problem or uncover progressive information for the environment in which the project lived, and thus garnered different datasets, tools used, and cleaning processes.

When asked if they could think of anything that would make any of the procedures easier, not much feedback was given. Out of the 8 subjects that were asked this question, 5 of them responded in the negative. The rest responded mostly with regard to more automation, such as automatic documentation similar to Javadoc, and automated cleaning procedures. However, the tone of their responses lacked much conviction, as if there was a part of them that did not believe such things would come to fruition, or would be very useful. Based on responses to other interview questions, there are certainly problems that are encountered by the subjects when executing certain procedures. Does the lack of feedback for this question imply that data users continue to work in a certain way simply out of habit?

4.3 Documentation

The most common problem discussed by the interview subjects was the absence of adequate documentation. This documentation includes data documentation, workflow documentation, and procedural documentation. Despite our asking several questions explicitly about documentation, the topic continued to surface in other questions throughout interviews.

The majority of the subjects did not document the workflow of their projects. The subjects that did were documenting the details of the procedures more than the actual workflow. Only two subjects explicitly documented workflows—one by writing a Word document listing the order and schedule of procedures to be run, and another involved a team of collaborators that contributed to a centralized document using a wiki inside Microsoft SharePoint. That centralized document provided just the high-level information, such as "1. Run this Python script (found here) 2. Run this SAS program (found there)...". Two subjects did not think they documented their workflows explicitly, but did record the order of programs to be run—one used DOS batch files that coded which scripts run in which order, and the other enumerated the file names to be run. Both of these subjects admitted that this was their only documentation on workflow, and was specific to the subjects themselves: "If someone wanted to go back and recreate my research... it would be very, very difficult."

On the other hand, two subjects said that the work required for a particular project was "in my head", and not documented. Those two subjects as well as another explicitly stated that they then had to document what was done from memory for the purposes of papers and progress reports. There is therefore a categorical distinction with research documentation: documenting in-progress, and documenting post hoc. Aside from a small amount of code commenting, the only documentation of the work done in a project by one of the subjects in transportation was weekly reports, which were done in the post hoc manner and only because they were required by a supervisor. This post hoc method of documenting actually causes more work at the end of a project, and since it is done by memory, a lot of detailed information could be forgotten and thus would be omitted in the documentation.

When asked if documenting workflows would be useful, the subjects unanimously said "Yes." However, it was clear by their responses that some subjects had procedural documentation in mind, rather than workflow documentation ("Yes, there are processes... that can, and should be... a simple task of pushing a button", "Yes—in particular the cleaning—definitely"). Surprisingly, when asked whether documentation of the procedures involved in a project would be useful, there were many direct "yes" responses, but two of the responses were not so direct. One subject stated that it depends on the skills of the other person, the complexity of the project, and who the documentation is for. The other subject stated, "For the person doing the work it might not be helpful, although it might save time later on (not having to document after the fact), but it would be helpful in the event where someone else would have to take over the job at a moment's notice. Of course, it wouldn't ever hurt, either." The latter subject reinforces the argument that documenting in-progress can save time in the long run.

Quite a lot of the subjects were mindful of the possibility that someone may have to take over their work or have to recreate their results in the future. This appeared to be a concern for the majority, and was something that they had thought about before ("Good documentation means that if I get hit by a bus, the next person

would be able to pick up the work and run with it with minimal effort"). Throughout the interviews the idea that someone in the future could redo or reuse their work was brought up at least 15 times among the subjects. Some documentation was done specifically for someone else to take over the work, such as using scripts and report appendices "for the purpose of someone else wanting to do the same thing." Other affirmative statements include, "...great documentation used to be written... The thought was that someone would take over, but no one did", and "Documentation is built with the idea that others can take over the work if need be." Most of the subjects, however, conceded that their documentation was likely insufficient for anyone else to take over or reproduce their work.

The subjects were also asked explicitly if they documented the details of the procedures that they performed for their projects. Two subjects answered with an admitted "no", but the others responded with a variety of what they considered documentation. The majority of the subjects referred to code comments as a form of documentation in response to this question as well as others. One of the subjects that writes code explained that each routine is numbered and has a description of what it does so that if the subject was hit by a truck, someone could take over based on the code and its comments. However, other subjects seldom commented code, commented roughly 50% of the time, or made only small comments targeted at themselves rather than others. Two subjects discussed the code itself and the scripts used as documentation. Five subjects mentioned handwritten notes—some of which were not retained or used later.

Although some documenting is occurring by most of the subjects, most of the documentation being produced is not sufficient for the use of others, despite the subjects unanimously praising the usefulness of documentation. One subject mentioned commenting code just because they had heard that it is a good programming practice. Another subject stated that documentation is attempted, but "most of the time I don't unless I'm asked to." Yet another subject said that documenting only seemed important when it was known before-hand that multiple persons were to be performing a task numerous times.

The lack of sufficient documentation could be attributed to the perception that stopping work to document is tedious and time consuming, and that getting results from the project were the main focus. Indeed, two subjects expressed this view: "[Documenting] is the tedious part [of the work]. Writing the code and doing the analysis is what we enjoy... we're always thinking about the results and the reports... I really don't look at it [in terms of someone else having to replicate the work], but I probably should." The other subject expressed the feeling of being too busy to document, and wanting to "just move on" rather than having to stop and document something. The latter subject said it would be helpful to have a tool that would automatically document certain tasks.

The general thought of the subjects regarding documentation seems to be summed up by one of the subject's statements: "Better documentation means more documentation."

4.4 Reuse of project artifacts

Questions were also asked about what barriers the subjects faced when reusing another researcher's datasets or any other artifact of that researcher's project. The responses from the subjects were overwhelmingly consistent, pointing to a lack of documentation. It was almost unanimous that it was imperative for them to know exactly what was done to the data upon receiving it. "Once [data] has been cleaned by someone else, you're not entirely sure if what they did was the right thing [unless there is good documentation]." Not only are the processes involved important in documentation, but some subjects noted that the semantics and coding schemes of the data were also of importance. It was also said to be useful to know why something wasn't done in a project, as well as those procedures that were done.

Of all the subjects that were asked what kind of documentation would be the most useful, most of them said that detailed documentation in English sentences would be ideal. "Code comments are good for giving the programmer hints, but not enough for someone else coming in and doing the [project]." Even more specific, two of the subjects said that documentation should first contain relevant information at a high level, followed by low-level, detailed documentation. Throughout the interviews, the subjects' responses regarding documentation alluded to the importance of documentation at all levels of granularity; code commenting, workflow specifications, procedure descriptions, or data definitions alone are not enough. Every aspect of a project must be documented sufficiently in order for reuse to be an option.

4.5 Other notable problems

Aside from the major problem of insufficient documentation, the rest of the problems that the subjects faced seemed to be more varied.

One recurring problem that one subject described as "not particularly challenging, but time consuming", is missing data. In a project focus-

ing on automobile crash predictions, the state's inventory databases only contained about 50% of what was needed to calibrate the data model from the national-level to the state-level, so the missing data had to be collected manually by looking at aerial photography and Google Street View. When using survey data in a project regarding an activity-based travel model, one subject recounted that many times questions are missing or simply weren't answered, so those instances have to be considered for removal before integration. Another subject explained that one huge dataset didn't consistently collect middle names, and thus made it difficult to integrate the data when trying to match applicants to the enrollment data from a community college system.

Inconsistent data plagued some of the subjects as well. Due to there being so many data sources involved in one subject's project (which was also a common theme), most often the datasets are not all in agreement and may even be logically inconsistent, and "this is very frustrating to work with." For example, when physical land data, land usage data, and regional employment data are joined together, it is found that there are some households where there are no buildings, too many households for the number of buildings, or there is employment data for a region that is known not to be true. "It's really hard to reconcile these things."

The subject that encountered problems with missing survey data also encountered inconsistent data in the same project; the surveys were often fraught with miscoding, logical inconsistencies, and "sometimes you just need to redefine things a bit—the way things were coded originally isn't useful, or potentially misleading, so there is a lot of recoding of data that goes on." For example, determining whether two or more individuals carpooled is sometimes obvious based on the surveys, but other times it was very ambiguous.

In a project whose focus was to obtain the number of licensed primary-care physicians in the state, one subject received a dataset containing physician names and addresses, but it was unknown whether these addresses referred to servicing locations, billing locations, or third-party locations. Furthermore, some of those addresses were found to be wrong when compared with other databases. "The pain is finding out the data is not reliable." One subject simply stated that the "main issue" is datasets that don't match.

Similar to inconsistent data is the problem of errors hiding within the data from human inaccuracies when entering and/or collecting the data. For example, the surveys in the activity-based travel project contained travel times that were not reliable at all—"people are bad at recording their own departure and arrival times, so that has to be taken into consideration during the analysis phase." Not only does this problem affect the analysis phase, but the cleaning process as well, as in one subject's project focusing on public bus collisions in a metropolitan area. In this case, the cleaning of the incident data—the data entered by dispatchers upon the notification of a bus collision—is critical, because most often the dispatcher is not accurate; the dispatcher often records the location of a collision as the nearest stop, when it actually happened two blocks away, for example. The data must therefore be coded correctly in accordance with operator reports, supervisor reports, and police reports. This coding correcting is all done manually. "There's no way to automate a program to be able to do that." The manual work consisted of sifting through 6000 records to see which were valid and which were not.

When observing some of the subject's projects work on a computer, we noticed that each subject worked with quite a lot of files. Moreover, several subjects proclaimed during the interviews that they use "Save As..." as an ad hoc back-up and versioning system. Managing this plethora of files and directories is not only cumbersome, but very specific to the individual. Each subject had their own style of organization and file naming scheme. This would make it difficult for another user to take over their work or reuse those files without any kind of documentation on the file structure (none of which had any), especially if they were to be used in a particular order.

Even though none of our subjects had a degree in computer science, only two did not mention having written any code for their work. However, just because they have written code does not mean they are seasoned programmers. One transportation subject on several occasions expressed a frustration stemming from inexperience with programming. This inexperience, which many data users possess, also causes more work and extra time. For example, this subject frequently had a computer science student code necessary procedures because the subject did not have the programming experience to write the code. Further, once the subject learned how to program, there was still enough of a lack of certain skills that caused more programming time than necessary due to copying and pasting sections of code and manually changing parameters; a more experienced programmer could have solved the same problem with only a fraction of the code and little time. Therefore, some training in appropriate programming languages and programming practices could reduce the amount of work required to perform a research project.

Other problems communicated by the subjects include having to contact agencies by phone in or-

der to get documentation or correct errors in data that the subjects themselves do not have ownership rights to, agencies changing the semantics of frequently used data, lack of automation for procedures, and procedures that are "mind-numbingly complex." All of the problems acknowledged by these interview subjects result in extra work, and thus taking more of the researcher's valuable time.

One other problem that is worth noting was brought up by one subject in the transportation field: there seems to be a conflict between the IT environment (which includes database designers and managers) and the users of the data. Databases are often designed poorly, because the designers do not know enough about which data is needed, and which datasets might be integrated—much to the frustration of the end users. "The IT people will tell you that the end users of data don't have a good appreciation of database development and management... and if you talk to the data users, they will tell you the IT people haven't got a clue about [which] data is relevant and [which] data is irrelevant. The reason... that we found a lot of frustration in the [transportation] industry is that you have essentially two camps that weren't communicating with each other very well." Databases were found in a particular project that had excellent architectures, but didn't provide very useful data.

Standardization of procedures and formats would solve a lot of the problems that the subjects mentioned. For example, if cleaning procedures within an organization, or even across a field of research, were standardized, recipients of data would know exactly how it was cleaned. Another example would be if all post-secondary educational institutions followed a standard data format (either at the state level or national level), so that their datasets can easily be integrated for analysis by local and federal agencies. However, this is most likely unrealistic.

5 Future Work

An idea for a tool that might help data users was conceived by one of the co-Principal Investigators of this project. The tool would be a documentation and file manager that keeps track of changes to data files, code files, and any other files relevant to a project, as well as make it easier for data users to document their work. The idea is based on Revision Control Software (RCS), which is a tool used during software development to help track and control the process of making changes to the source code in the midst of many concurrent developers. RCS allows users to revert files back to previous versions, as well as see who made changes to certain lines of code at any given time. In addition, there is an interface for the users to enter any comments about what changes are about to be applied, or *committed*, to the code.

With this in mind, we included a section of questions in the interviews regarding RCS, and how it might apply to the subjects and their work. Half of the subjects had heard of RCS, but only one could give some sort of description of what it was. After explaining RCS to the subjects, we asked them if they thought it would be useful to them. The subjects almost unanimously said "yes"—only one said that it didn't sound useful, but could see it being useful to others. Many of the subjects who answered affirmatively gave enthusiastic comments, such as "definitely", "that would be good", "would be very useful", and "it does sound interesting." Several subjects observed that a tool like RCS would provide a better backup system than frequent use of "Save As...". In addition, one subject stated that it would be "*really* useful if [RCS]" could encompass any and all tools involved in projects—not just a specific subset of tools.

We then drilled down further and asked the subjects about the usefulness of four specific features and benefits of RCS: the ability to recover files to a previous state, managing separate *branches* for concurrent work by multiple users, the ability to query for documentation of what was done and why, and the propensity to motivate its users to document more. With the exception of the branching feature for concurrent users, all but one of the subjects that were asked these questions gave a spirited response that these features would definitely be useful to them. With regard to the ability to recover previous states of files and/or data, one subject noted that this can save a lot of time by eliminating the need to start over from scratch in cases when errors have been made somewhere along the workflow, for example. Although the branching feature for concurrent work by multiple researchers on the same project still had a majority of the subjects expressing its usefulness, it did spawn less enthusiasm. Those who responded negatively or neutrally reasoned that the work they do is strictly independent; two of those subjects added the caveat that they could see it being useful to those that did need to collaborate on a project. The one subject that did not find RCS useful reasoned as such because he felt that the methods and tools the subject used were sufficient, and did not need the features RCS has to offer.

Based on these interviews—particularly the questions regarding Revision Control Software—we created a mock-up of a tool that could make documentation and file management easier for data users. This tool, called WHIM, Work

History Information Manager, provides the researcher with an unobtrusive interface with which to document the work done for a project. The details of this tool is beyond the scope of this paper, but the next step for this research team is to implement a working prototype of WHIM.

A presentation was created to show how a data user might use WHIM using a mock-up user interface. The scenario of the faux data user in the presentation was loosely based on one of the subject's projects in the transportation field. The presentation was then shown to four of the interview subjects (all of whom are in transportation). The subjects were unanimous in their opinion of what they had witnessed; the subjects all thought very highly of the tool, and expressed interest in using it. Once the presentation ended, one subject replied, "When can I get it?"

6 Conclusion

We set out to discover what data users specifically do in their projects, what problems they encounter when working with data, if and how they document their work, and how much value they place in documentation of projects. By interviewing ten data users—seven of whom belong to the transportation area of research and analysis—we learned that the subjects 1) encountered poor documentation in project artifacts that were received from external sources and thus had to do more work, 2) did not adequately document their own work, 3) unanimously agree that documentation is very important and could remove a lot the barriers they face, and 4) find the idea of revision control software useful.

It was fairly clear from amount of overlap between these interviews that the main reason for the inability, or hesitation, to reuse a project—either in whole or in part—is a substantial lack of proper documentation. We have created a mock-up for a tool called WHIM that is based on Revision Control Software that could greatly increase the amount of documentation produced by data users, provide an easy-to-use interface for documenting the work done in projects, and give data users the ability to easily go back to previous states of files and data.

Before one of our interviews, we explained this research project to the subject. In response, the subject stated very succinctly, "Anything that will make documenting easier would be ideal."

References

[1] Heather Alyce Piwowar, *Foundational Studies for Measuring the Impact, Prevalence, and Patterns of Publicly Sharing Biomedical Research Data*, University of Pittsburgh, 2010.

[2] Ann S. Zimmerman, *Data Sharing and Secondary Use of Scientific Data: Experiences of Ecologists*, University of Michigan, 2003.

[3] Sergio Cozzetti B. de Souza, Nicolas Anquetil, Káthia M. de Oliveira, *A Study of the Documentation Essential to Software Maintenance*, in: Proceedings of the 23^{rd} International Conference on Design of Communication (ACM SIGDOC '05), Coventry, UK, 2005, pp. 68-75.

[4] Mary Vardigan, Pascal Heus, Wendy Thomas, *Data Documentation Initiative: Toward a Standard for the Social Sciences*, The International Journal of Digital Curation 1 (3) (2008) 107-113. July.

[5] Alice Robbin, *Managing Information Access Through Documentation of the Data Base: Characterizing Social Science Data Base Text Documentation as a Minimal Information Management System*, ACM SIGSOC Bulletin 6 (2-3) (1974-1974) 56-68.

[6] Aída Gándara, George Chin Jr., Paulo Pinheiro da Silva, Signe White, Chandrika Sivaramakrishnan, Terence Critchlow, *Knowledge Annotations in Scientific Workflows: An Implementation in Kepler*, in: Proceedings of the 23^{rd} International Conference on Statistical and Scientific Database Management (SSDBM '11), Portland, OR, USA, 2011, pp. 189-206.

[7] Laura Wynholds, David S. Fearon, Jr., Christine L. Borgman, Sharon Traweek, *When Use Cases are Not Useful: Data Practices, Astronomy, and Digital Libraries*, in: Proceedings of the 11^{th} International ACM/IEEE Joint Conference on Digital Libraries (JCDL '11), Ottawa, Canada, 2011, 383-386.

[8] Sam Uselton, *Multi-source Data Analysis Challenges*, in: Proceedings of the Conference on Visualization (VIS '98), Durham, North Carolina, USA, 1998, 501-504.