White Paper on Ownership of data, testbeds and artifacts

Vic Basili, Marv Zelkowitz, Dag Sjøberg, Philip Johnson, Tony Cowling

The goal of this white paper is to provide some guidance and discussion on the following questions:

- How should data, testbeds, and artifacts be shared?
- What limits should be placed on who can use it and how? What credit should be given? How does one limit potential misuse?
- What is the appropriate way to give credit to the organization that spent the time and money collecting the data, developing the testbed, building the artifact?
- Once shared, who owns the combined asset and how should it be shared?

Introduction

A great deal of what is involved in empirical study involves the generation of data, artifacts, and the use of experimental testbeds of various kinds. These involve a fair amount of work on the part of the group who developed them. So, on one hand, the owners/developers of these items would like to share them with the community for a number of reasons, e.g., to contribute to the replication of experiments, to allow meta-analysis, to get more people involved in the empirical research. On the other hand, they would like to control the use of these artifacts so they know how they are being used, so they are not misused, so that the results of any work using these items is available to them as well. It is also worth noting that it costs money to maintain these items, interact with the requesters by answering questions, track their evolution, etc.

The purpose of this white paper is to begin a discussion within the community on how such items could/should be shared that is equitable to all parties.

One end of the spectrum is to allow anyone who wants to use the items, use them at their own risk, with whatever problems that occur. Usually this incurs some cost on the part of the owner but if he/she is willing to incur the cost, it is up to him/her. The other end of the spectrum is to limit us to only those groups that are working directly with the original developer of the items.

Another problem has to do specifically with artifacts. One evolutions. [DS1]Artifacts were evolved and characterized for a specific environment and may not have the same meaning/relevance in other contexts. Thus several benefits were not obtained for the community.

Background

What follows are several examples of data, testbeds and artifacts that were made available and how they shared data and what problems they had. The examples include the NASA SEL experience base, the set of laboratory manuals for the reading experiments ~~form~~ from the UMD Experimental software engineering group, the CeBASE experience base of empirical results, and the Fraunhofer/UMD TSAFE experimental

environment, ~~and~~the Hackystat databases and the projects conducted within the Sheffield Software Engineering Observatory.~~.~~ (VB: More experience should be included, e.g., the Hackystat experience, the Observatory Project)

Each study tries to answer the following questions:

What was the goal of the project?
What kinds of data were collected and/or artifacts developed?
How many users were there, if known?
What are the primary problems associated with the data ownership and how were they dealt with?

1. NASA SEL database. The NASA Software Engineering Laboratory (SEL), begun in 1976, was aimed at studying software development for Ground Support Software at NASA/GSFC with the goal of improving the quality of the software developed []. It collected data resources expended, changes and defects, product characteristics, and processes applied as well the conformance to those process on several hundred projects. Various project characteristics were also collected, e.g., context variables. The artifacts developed where real ground support systems. Models of cost, schedule, and quality were built using this data.

For years the data was given away freely and all artifacts were shared with whoever asked for them. It is therefore hard to estimate how many people used the data as it was available form NASA and from the Rome Air Development Center repository. Problems ~~The negative aspects~~ from the owner's point of view were (1) the data was often misused and misinterpreted because there were not a sufficient number of context variables, (2) the results of the analysis were often not known/shown to the original data owners, (3) the data owners often had to spend time in organizing and making the data available to requesters and answering questions, and (4) the opportunities for feedback and meta-analysis were often lost. Misuse of the data (item 1) was also a problem for the requestors as well as the community since questionable results were published. Late in the life cycle of the project, it was recommended that anyone who wanted to use the data, should spend some time at the SEL, interacting and understanding the nature of the data.

2. The Set of Software Analysis Projects. In the mid-1080's, an experiment was run to evaluate the effectiveness of reading as a defect detection technology. The first of these experiments was aimed at evaluating the effects of reading vis-à-vis testing on code. Later studies involved the evolution of reading techniques based upon empirical evidence from applying the techniques in a variety of environments. Major foci included the development of techniques for reading requirements documents and object oriented designs. To run these studies several ~~Another example is the use of~~ artifacts such as fault seeded code documents or requirements documents that have been developed and data collected on the defects detected by various techniques, or versions of techniques, applied to the artifacts. The reuse of ~~for one set of experiments, being used in another. This~~ has the advantage of allowing a form of replication of the original study by multiple sources and promotes empirical work by making it easier for researchers to get started in

empirical work. For example, several groups replicated the original reading vs. testing experiment [Basili&Selby], as well as the perspective-based reading experiments [Basili, et al.]. However, our experience has been that it is difficult to replicate the original work without a certain amount of implicit knowledge from the original researchers [EMSE paper on tacit knowledge]. This involves time and money in sharing results but the consequence, if this information is not shared, is that there will be badly replicated projects or non-combinable results. It is also possible that the original researchers will not even know about the follow on work and therefore cannot help or critique the results of follow-on studies. A good counter example has been the Reader's Project [Reader's project], a collaboration among the original developers of a set of reading experiments [ ] at the University of Maryland and a group of researchers in Brazil, where experiments were replicated, meta-analysis performed, and recognition of the problems with laboratory manuals identified, e.g., the kind of tacit information that is not in the laboratory manual. Here joint funding was received to support the work (from both countries). The results were very successful replication of experiments and a great deal of learning on all sides about replication, improvement of the artifacts, laboratory manuals, etc.

3. CeBASE, (Center for Empirically-based Software Engineering). CeBASE was an NSF sponsored project with the role of acting as a repository of "experience" associated with the application of various software development methods, techniques in order to create hypotheses, qualitative and quantitative models of, and any other form in aggregated experiences. The goal of the original project was specifically to create a shared repository of empirical information on the effects of applying a variety of techniques, methods and life-cycle models. Areas of interest for sharing have been defect detection methods, COTS developments, and Agile methods. The emphasis in CeBASE has been to get the collaborators to provide "experience" in the form of qualitative information on the effects of applying technologies.

The approach has been to ask CeBASE collaborators to sign up, so they are publicly affiliated with the project. However, there is no monitoring so non-affiliated people can use the experience base and members can use it without the owners knowing about it and providing feedback to the experience base. Other issues are:

Who pays for the cost of maintaining the experience base?
Who has access to analyze and synthesize and create new knowledge?
How do we assure they provide quality data, analysis, and new knowledge?

The original experience base maintenance was supported by the NSF grant, but once the grant ended, maintenance has become the responsibility of one of the CeBASE participants with the need to keep the EB alive. We have applied for infrastructure funding from NSF but that is usually allocated to hardware. Without external funding, maintenance is at a minimum.

4. **The High Dependability Computing Project**. In the High Dependability Computing Project (HDCP) has the aim of defining models of dependability and assessing the ability of new research techniques to support the development of highly

dependable systems for NASA. Example applications include systems such as the Mars Scientific Laboratory and the Earth Observatory System Data Information System. In order to reduce the risk of applying these technologies to live systems before they are demonstrated effective, we are building experimental environments consisting of sets of testbeds (SCROVER, TSAFE) to be used to compare the effectiveness of various techniques for improving software dependability in isolation and in combination. The testbed artifacts include requirements (functional and non-functional), design artifacts, test cases, defect seeded versions of all the artifacts, results of applications for various techniques, etc. The artifacts will evolve over time. The testbeds are currently being used by a small set of researchers to test out their techniques. At the moment, there is more control as the experimental environment users need to interact at some level with the testbed developers. But should the testbed be released on the web and what happens when that happens? Also, the testbed needs to evolve based upon the needs of the different technologies to have different artifacts for analysis.

Who will pay for this?
How will the testbed and the results be maintained?
How should the artifact creators be acknowledged?

5. Open source development. There are many commercially competitive products now being developed via the open source route. Some organization of unpaid workers (either at one company or a loose collection of web-enabled individuals) control updating of the official source library, but anyone is able to copy anything from the library and propose changes to any artifact in the library. There are rules that anything modified from the library is part of the open source product and is freely available to anyone. Products such as Linux, Eclipse environment, and the Apache web server are all open source. Companies make money by providing these basic products and then sell proprietary enhancements to the basic product (e.g., Red Hat Linux). Can the open source mechanism be used to maintain and enhance software engineering measurement and experimentation artifacts? Is there a commercial market that will allow the free-based open source mechanism to work?

6. Hackystat. The Hackystat project explores automated collection and analysis of software engineering metrics. In Hackystat, sensors are attached to individual development tools, which unobtrusively collect data about product and process and send them to a central server where analyses over these data can be performed. Examples of sensor data include: activities (compilation, file editing, etc.) within an editor (such as Emacs, Eclipse, JBuilder, etc.), invocation of unit tests and their results (using a test framework such as JUnit), size/complexity of the system (in LOC, methods, classes, operators, etc. using a tool such as LOCC), configuration management events (such as commits and lines added/deleted using a tool such as CVS), test case coverage (using a tool such as JBlanket), software review process and products (time spent doing review, issues generated, etc. using a tool such as Jupiter), and defect management (using a tool such as Jira). From this raw sensor data, higher level abstractions can be built regarding the trajectory of development, and analyses can be performed to look for trends in sensor data and co-variance over time.

Over 150 users have sent data to the public Hackystat server since the summer of 2001 when it was first made available. Approximately 1.5 GB of raw sensor data, representing over 10,000 person-days of development currently exist on this server.

To date, the use of this data has been ~~focussed~~focused on education (teaching students how to collect and interpret their own process and product metrics) and usability evaluation of the Hackystat framework itself (using the data to better understand important areas for improvement).

A primary, long term goal of the Hackystat Project is to convert this metric repository from a representation that is useful only to the users who generated their data to a representation that provides information of general utility to the software development community. There are two immediate challenges in ~~acheiving~~achieving this goal related to this white paper. The first is privacy: what kind of 'data scrubbing' process should be performed on the data in order to allow its publication without revealing identifying details? The second challenge stands in direct opposition to the first: without the addition of some kind of 'demographic' information about the developers, the product under development, and the context (process), it is unclear how the larger community could obtain value from the Hackystat data. In summary, it appears that we must simultaneously add and subtract information from the current Hackystat database to make it usefully publishable.

7. The Sheffield Software Engineering Observatory. This provides an environment within which student teams carry out various forms of software development projects, but for real clients (ie external to the university), and under conditions that are as industrially realistic as is possible within a university context. For the "software hut" projects the students are second-year undergraduates (within the UK pattern of three-year honours bachelors' degrees), while for the "Maxi" and "Genesys Solutions" projects the students are on masters' courses. In the software hut and Maxi projects each client has a number of teams working with them on the same business scenario, and these teams are each competing to build a system that will best meet the client's needs. This set of teams working with one client is divided into two or more groups, with each group using a different methodology, so as to allow experimental comparison of complete methodologies, where the focus so far has been on comparing traditional methodologies with agile ones (viz XP and variants of it) [**can provide refs**]. By contrast, Genesys Solutions operates like a conventional software house [**can provide a ref to its website**], which is run by the students under the oversight of the academic staff, and it focuses on the use of XP. Here each project just has one team working on it, but typically the projects are larger and involve more development effort than either the software hut or Maxi projects, thus allowing detailed observations to be made of how XP actually works for industrial–scale projects.

Three main kinds of data are collected within the Observatory from these projects. Firstly, a complete repository of project artifacts has been built up: management plans, requirements documents, design documents, test plans, test harnesses, code, etc. For each

project these are also supplemented by documents that reflect its role in the student's courses, so that (for instance) they are required to write reports evaluating their own work and that of their teams, and the clients are also asked to provide evaluations of the systems that the student teams have developed. Secondly, basic quantitative data has been collected during the development processes, for instance on the amount of time that each person has spent on different activities, although hitherto we have not been able to collect this in as much detail as a system like Hackystat would allow: this is a development that we hope to make soon. Thirdly, researchers have observed teams while they work on the various projects, and in some cases have also worked with them, for instance as XP coaches. From these observations they have collected qualitative data about the interactions between team members and their attitudes to the work that they have been doing, supplemented by quantitative data from tests such as the MBTI, Warr's well-being measure, PANAS and the workgroup cohesion measure [**can provide refs for these**].

In terms of this white paper, the two key issues surrounding the release of raw data are those of client privacy and developer privacy. For each client the raw data contains much information about the nature of their business that could be commercially sensitive, and while only a few of them require us formally to sign non-disclosure agreements, we do require all the students to sign such agreements. Thus, any release of raw data would have to be on the basis of the researchers to whom it was released signing a similar agreement, and this would have to require that it applied transitively to any situation in which they in turn were going to release some of the data. Similarly, for each student developer the raw data contains information not just about the work that they carried out, but about aspects of their involvement in it that they might well wish to keep private from those outside the team in which they had been working. As far as the observational data is concerned this is dealt with partly by anonymising it before it goes into the database, but in the project artifacts and the quantitative data the individuals are often identified by name, and it would require considerable effort to anonymise it before release.

Guidelines

Based upon the above experiences, we might recommend the following guidelines for sharing:

As a minimum, the requester of the use of an artifact or data should (1) officially request **permission** to use of the items, e.g., write a white paper what they plan to do, ask permission, … (2) **credit** the original developer with the work involved, (what should they reference?) (3) offer the opportunity for **collaboration** (where real collaboration is expected), (4) provide **feedback** on the results of use as well as problems with using the artifact or data. There are also issues concerning the **protection** of the data and artifacts like privacy, safety, etc. that need to be considered. For example, the data was most likely collected from people who were assured anonymity – how should this be handled?

The following are views about permission, credit, collaboration, feedback, and protection.

Permission

Does one have to request permission to use the material? Is it simply publicly available? What should be the rules?

How does one provide some form of controlled access to the items. There might be a request to use the artifact with a commitment to provide feedback after or during use (method, results, other data) and reference the items in all work using them. We can actually restrict access by requiring that the requestor write a one page proposal to the data owner.[DS2] [DS3]Then the item can be used:

- Freely, in the public domain
- With a license
- With a service fee for use (by industry) to help maintain the data

Credit

How should the original group gathering the data or developing the artifact be given credit? What would be the rewards for the artifact/data owner? What credit should be given?

The type of credit is related to the amount of interaction. If there is an interaction, depending on the level, co-authorship may be of value. If it is used without the support of the data owner, some credit should still be given, e.g., acknowledge and reference the data owner. Thus, if the requestor uses it but the owner is not interested in working on the project, the minimal expectation is a reference and an acknowledgement. (We need to think about how that reference should be made, e.g., the paper that used them or some independent item where the artifact itself exists a reference.) [DS4]It is also possible that some form of "associated" co-authorship might be appropriate.

(A side issue, independent of external interactions, is internal interactions. Experiments require a team of people (I am suspicious of anyone who runs an experiment by themselves.) How is authorship of such work decided? Does everyone who contributes get on the author list? What should be the order of authors? Can we provide some form of guideline here? For example we have been criticized for having too many authors on a paper. This is common practice in experimental physics (everyone from the builder of the instrument to the original experimenter can be an author of a paper) but not accepted practice in computer science. Is this something we should fight[DS5]?)[DS6]

Collaboration

In general it is recommended that the requestor keep the option open of collaboration on the work. Some options are suggested above. The Reader's project has been an excellent example of collaboration but required funding on both sides and the collaboration is no longer funded. Funding agencies are often looking for "new" ideas and so it is often difficult to be funded for a continuing operation. What options are there for funding collaborations? (If collaboration is not desired by the owner of the artifacts, what are the rights of the requestor? It is probably too strong to require collaboration as a requirement for any requestor.)

Feedback

At the very least, by using the permission or request item, there is a sense that the originator of the materials knows of someone is using their materials. However, some form of feedback can act as payment, i.e., updated versions of artifacts, data so it can be used in some form of meta-analysis, some indication of the effectiveness of technology on the experimental environment. A lot depends on what the originators' fell[???] will be useful. Maybe this can be part of the original agreement.

A related issue is assuring that the quality of the data, analysis, and new knowledge being returned to the originator is acceptable. (to whom? Not sure you can say anything here.)

Protection

There are a large number of issues here. How does one limit potential misuse? How does one support potential aggregation and assure it is a valid aggregation. How does one deal with proprietary data? Even when data is not proprietary, how does one assure anonymity of the data.

What is required on the originator's side? Should they be allowed/required to review results before a paper is submitted for external publication. (Too strong.) Should there be some form of permission required by reviewers? Who has the rights to analyze and synthesize and create new knowledge based upon the combined results of multiple studies? Again here, how is credit given, authorship determined?

How does one limit potential misuse?
Lay out some options for collaboration

Maintenance

One major practical issue is the maintenance of the experience base of data or artifacts. Who pays for the cost of maintaining the experience base? The CeBASE project has appealed to NSF to support the infrastructure but it is not clear how this requires will be received. (There are only 3 possibilities here: (1) Owner of the data, (2) Users via a licensing fee, (3) Everyone via an open source arrangement. (1) won't work since few have such resources, (2) will limit use – researchers won't generally pay. (3) maybe, but difficult as stated previously.)

Conclusion

The ISERN community has the need to raise the problems associated with experimentation and the sharing of materials and has the visibility in the community to be listened to. We can use EMSE as a potential sounding board for opinions.

~~Old Comments from the ISERN Workshop:~~
~~Mike Holcombe and Tony Cowling~~

~~Open Source Model~~

•We would expect to make the data available for others to use, and we would expect them to identify clearly the source of any data that they did use.
•But, if they wanted to use our data then it would have to be on the basis that they would be assumed to know what they were doing with it, so that beyond the normal conventions of documenting it properly, we would not expect to have to make any commitment at all to providing any particular assistance with using it.
I hope that this remote contribution may be of some help, and look forward to finding out what the outcome of the discussion is.

Hackystat Architecture

How should data, testbeds, and artifacts, be shared?
•What limits should be placed on data sharing, use of artifacts?
–An written agreement should be reached between both parties
–Various options should be considered
–But at the least the researchers must provide feedback (method, results, other data)
•How might the agreement proceed?
–Recommend that the requestor write a one page proposal to the data/artifact owner
–Keep the options open for collaboration on the work, assuming both parties are interested in working on it
–Need to define the win-win incentives
–Data sharing and artifact use should take place in the context of a larger project
–Use citations with the list of developers/creators, and acknowledge the data, lab manual, artifact
–(This can encourage collaboration – an ISERN goal)
–Need a variety of options from co-authorship to, giving the data, to not giving the data