Measuring Reproducibility in Computer Systems Research

University of Arizona TR 13-03

Christian Collberg collberg@gmail.com Todd Proebsting proebsting@gmail.com
Gina Moraila gmoraila@email.arizona.edu
Akash Shankaran akashs@email.arizona.edu
Zuoming Shi zuomingshi@email.arizona.edu
Alex M Warren amwarren@email.arizona.edu

December 10, 2013

Abstract

We describe a study into the willingness of Computer Systems researchers to share their code and data. We also propose a novel *sharing specification* scheme that will require researchers to specify the level of reproducibility that reviewers and readers can assume from a paper either submitted for publication, or published.

1 Introduction

Reproducibility is a cornerstone of the scientific process: only if my colleagues can reproduce my work should they trust its veracity. In the wet sciences reproducing someone's experiment can be difficult, often involving expensive laboratory equipment and elaborate processes. In applied Computer Science, however, this should seldom be so: unless esoteric hardware were used, reproducing the work published in a systems conference or journal should be as simple as going to the authors' website, downloading their code and data, typing "make," and seeing if the results correspond to the published ones.

Anecdotally, this is not so. In our own experience we have multiple times contacted the authors of a particularly interesting paper, asked for source code, either to be met with a negative response or stone cold silence. We present two such anecdotes in Appendix A. It is our belief, and the beliefs of many [5, 13, 19], that such unwillingness or inability to share ones work with fellow researchers hampers the progress of science and leads to needless replication of work and the publication of potentially flawed results.

It should be clear that in order for Computer Science research to be reproducible, several hurdles have to be cleared: the source code and test case data have to be available, the code has to build, the execution environment has to be replicated, the code itself has to run to completion, and accurate measurements (with respect to performance or other metrics) have to be collected. In this paper we concentrate on the two most basic of these requirements: *is the source code available, and does it build?*

To investigate the extent to which Computer Science researchers are willing to share their code and data, and the extent to which this code will actually build with reasonable effort, during the spring and summer of 2013 we performed the following study. We downloaded 613 papers from the latest incarnations of eight ACM conferences (ASPLOS'12, CCS'12, OOPSLA'12, OSDI'12, PLDI'12, SIGMOD'12, SOSP'11, VLDB'12) and five journals (TACO'9, TISSEC'15, TOCS'30,

TODS'37, TOPLAS'34), all with a practical orientation. For each paper we determined whether the published results appeared to be backed by source code or whether they were purely theoretical. Next, we examined each non-theoretical paper to see whether it contained a link to downloadable code. If not, we examined the authors' websites, did a web search, examined popular code repositories such as `github` and `sourceforge`, to see if the relevant code could be found. In a final attempt, we emailed the authors of each paper for which code could not be found, asking them to direct us to the location of the source. In cases when code was eventually recovered, we also attempted to build and execute it. At this point we stopped—we did not go as far as to attempt to verify the correctness of the published results. The details of how this study was conducted can be found in Section 3.

The results of our study can be found in Section 4.

Related work, described in Section 2, falls in three categories: some authors decry the current state of affairs in reproducibility and present a call to arms for more ethical science; some authors describe the steps that need to be taken in order to produce research that is truly reproducible; and some authors describe tools or websites that support the publication of reproducible research. Our recommendations (detailed in Section 5) are more modest. We recognize that, as a discipline, we are a long way away from producing research that is always, and completely, reproducible. But, in the interim, we can require authors to conscientiously inform us of their intent with respect to publishing their code and data: will it be made available, how will it be made available, in what form will it be made available, on what platforms will it run, what support will be provided, etc. We believe this information should be provided by the authors *when their work is submitted for publication*, allowing reviewers to account for the expected level of reproducibility in their decision to accept or reject.

Thus, this paper makes the following contributions: we describe a rigorous process for determining the level of reproducibility in Computer Systems research (Section 3), we describe the outcome of a reproducibility study of 613 papers published in top ACM conferences and journals (Section 4), and we make a recommendation for adding *sharing specifications* to publications that represent a contract between authors and their readers as to the promised level of reproducibility (Section 5).

All the code and data from this project can be downloaded from our website[1].

2 Related Work

In this section we review some of the prior work related to reproducibility. In particular, we examine previous studies into reproducibility, tools for supporting reproducible research, and legal and licensing frameworks.

2.1 Empirical Studies into Reproducibility

In this paper we conduct a comprehensive study into the willingness of Computer Systems researchers to share their code with colleagues. Two similar studies have been conducted in the past. In 2007 Kovacevic [10] examined 15 papers published in the *IEEE Transactions on Image Processing*. She read the papers and rated them on how well algorithms were explained and whether code and data were available. She found that while all algorithms had proofs, none had code available, and 33% had data available.

Vandewalle et al. [20] distinguish 6 levels of reproducibility, starting at level 5 where *"The results can be easily reproduced by an independent researcher with at most 15 min of user effort,*

[1] `http://reproducibility.cs.arizona.edu`.

requiring only standard, freely available tools (C compiler, etc.)", to level 4 where *"proprietary source packages (MATLAB, etc.)"* are required, down to level 0 where *"The results cannot be reproduced by an independent researcher."* They then repeated the study from Kovacevic [10] with a larger sample size, all the 134 papers published in *IEEE Transactions on Image Processing* in 2004. Each paper was scored on its reproducibility by two or three reviewers. They found that *"code (9%) and data (33%) are only available online in a minority of the cases, with data being available more often thanks to the frequent use of standard image data sets, such as Lena."*

While the Kovacevic and Vandewalle studies consider many aspects of reproducibility, such as whether algorithms (subjectively) are described clearly enough to be independently implemented, our study focuses entirely on whether code and data are made available. We also consider a larger sample size (613 papers) and a larger number of journals and conferences (13). We furthermore attempt to actually build and execute the source and contact the authors of code not found on public repositories to see if they would provide it when asked directly.

Stodden [17] reports on a survey of 638 registrants at the NIPS machine learning conference as to their willingness to reveal code, data, and ideas. The survey showed that 74% were willing to share post-publication code, and 67% post-publication data. In contrast, Stodden found that "30% of respondents shared some code and 20% shared some data on their own websites." The most common reasons for not sharing code were found to be "The time it takes to clean up and document for release," "Dealing with questions from users about the code," "The possibility that your code may be used without citation," "The possibility of patents, or other IP constraints," and "Competitors may get an advantage."

2.2 Public Repositories of Code and Data

Stodden et al.'s [18] RunMyCode web site[2] provides "computational infrastructure" allowing authors to publish the code and data related to their papers. In addition to making code and data available for download, the site also allows users to execute the code in the cloud. Unfortunately, the *"if you build it they will come"* paradigm does not always work: so far (July 2012) only 80 papers appear to have been registered on the site, none of them in the Computer and Information Sciences category.

Gavish and Donoho [6] describe a system whereby data and scripts are stored on (journal) web repositories, and every computational result (tables, figures, charts, datasets, etc.) is given a *Verifiable Result Identifier* (VRI). VRIs are included directly into the article source, allowing reviewers and readers to interact with the scripts and data (verifying results, examine data, re-execute computations) without having to explicitly download anything. Gorp and Mazanek [8] describe a similar system, called SHARE, a web site that allows authors to share remote virtual machines that contain a complete execution environment, code, data, and the paper text itself. Readers can click on links in a paper to connect to these virtual machines where they can interact with computational artifacts. Paper Mâché [3] is yet another system that supports the *executable paper*[3] concept, also using virtual machine technology to replicate the execution environment in which scientific computations were run.

CARMEN [2] (Code, Analysis, Repository and Modelling for e-Neuroscience) is a web based portal which allows users to share code and data in neuroscience.

Koop et al. [9] describe a provenance-based system that captures workflows, i.e. exact histories of how data was captured, experiments performed, and results were generated. Similarly to the systems described above, clicking on a plot in a paper will open up the original workflow from which

[2] `http://RunMyCode.org`.

[3] `http://www.executablepapers.com`.

it was generated. Additionally, the system described by Koop et al. integrates version control so that authors can revert to previous versions of data and code.

Muller et al.'s [1] A-R-E (Author-Review-Execute) environment provides similar facilities as in the systems described above, but also provides anonymization during the review process. The idea is to place an anonymizing proxy between the reviewer and the author's web service that provides access to data and code.

2.3 Processes, Procedures, and Publication Practices

Schwab et al. [14] describe how to use `make` and related tools along with naming conventions for input and generated files to ensure that readers as well as authors can reproduce computations.

Lepton [11] is a literate programming system that makes papers self-contained, by integrating code and data within the paper itself. Gentlman [7] presents a similar idea based on literate programming, specifically for the R language.

Limare [12] describes experiences gained from editing the journal *Image Processing On Line* where "Each article contains a text describing an algorithm and source code, with an online demonstration facility and an archive of online experiments. The text and source code are peer-reviewed and the demonstration is controlled." In particular, he points out the risk of obsolescence of code stored in the repository, as programming languages and libraries change over time.

2.4 Licensing Frameworks

Stodden [15, 16] has proposed "The Open Research License" to incentivize researchers to share their code and data without fear that their work will be expropriated by others, without proper attribution. Her licensing scheme combines the Creative Commons license[4] for the media portion of the research and the BSD[5] license for the code portion. The hope is that universal use of such licensing schemes will "encourage reproducible scientific investigation, facilitate greater collaboration, and promote engagement of the larger community in scientific learning and discovery" by ensuring "each scientist is attributed for only the work he or she has created" [15].

3 Process

Figure 1 shows the process by which our study was conducted. In an initial step, each paper from the selected conferences and journals were downloaded from the ACM Digital Library or directly from the conference web site. The papers were next scanned manually by a team of undergraduate and graduate research assistants to determine if the results reported on were in any way backed by source code. Purely theoretical papers or papers that relied on hardware not available to us were not considered further.

An attempt was next made to find links to any source code from the article body and bibliography. If no link was found, a web search was carried out via Google and popular code repository sites such as GitHub, Google Code, and SourceForge. Search queries combined the tool name and fields such as institute name, author name, and other keywords. If the institution or the authors had public web pages related to the project, those pages were considered as well.

In order to ensure accurate results, each paper was reviewed by two research assistants. In addition, random papers were further selected to be reviewed by a professor, to monitor the quality

[4] `http://creativecommons.org/licenses`.

[5] `http://opensource.org/licenses/BSD-3-Clause`.

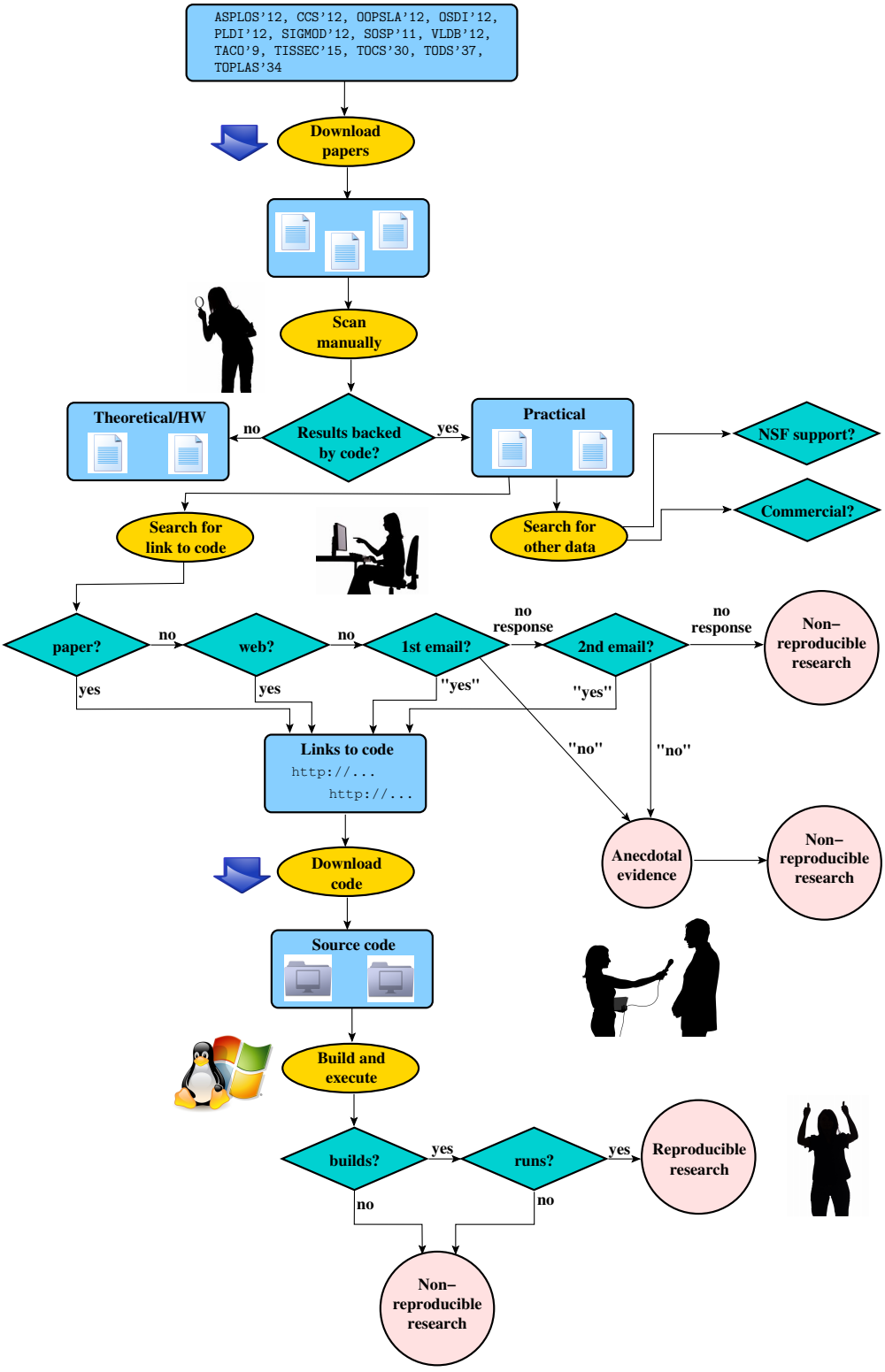


Figure 1: Process by which the study was performed.

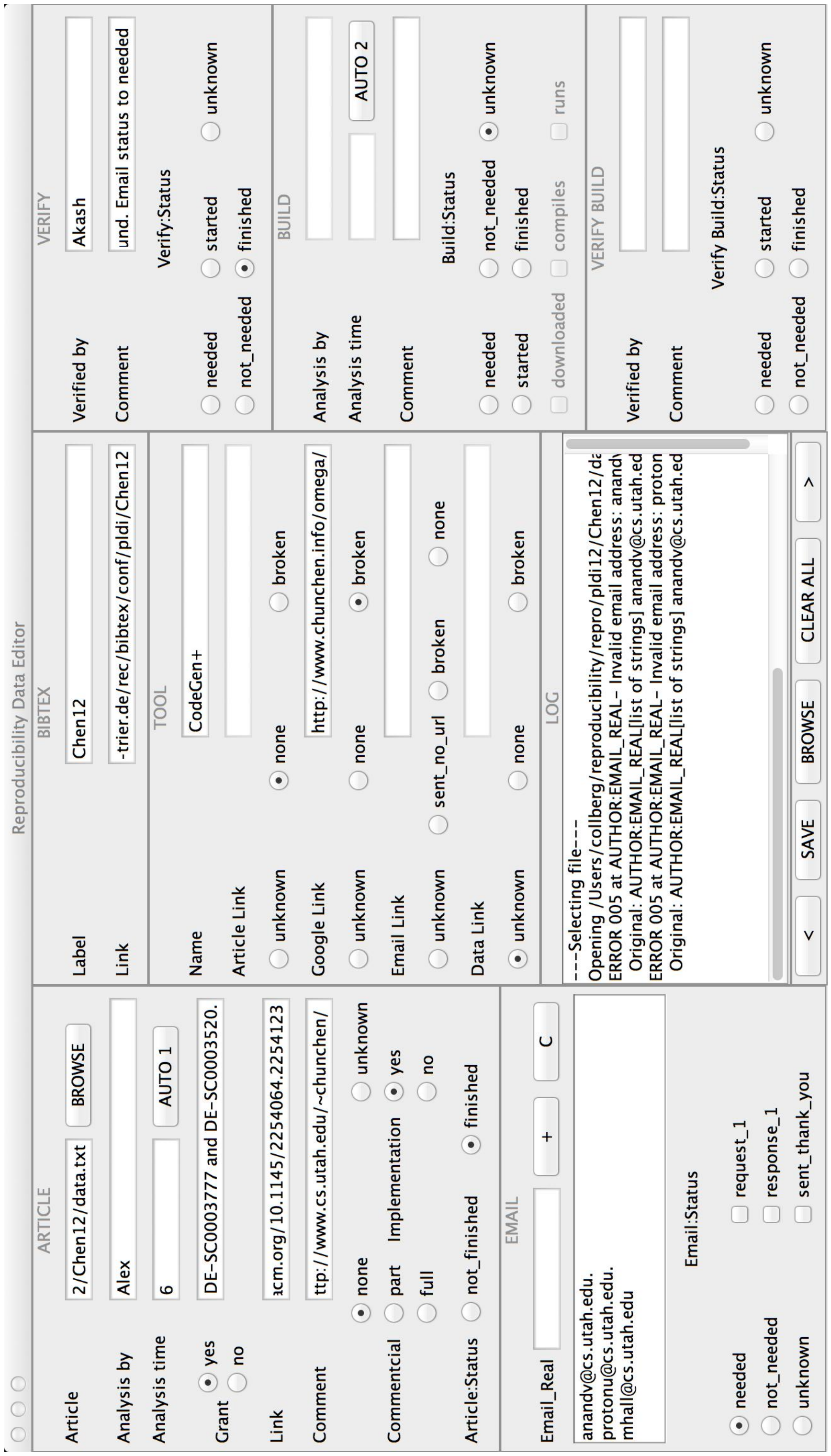


Figure 2: Editor to enter data during the study.

```
From: Christian Collberg <ccollberg@gmail.com>
To: first-or-corresponding-author
Cc: remaining-authors
Subject: Your conference-name paper

Dear Dr. first-or-corresponding-author,

I've been looking at your conference-name paper
   paper-title
and would like to try out the implementation. However,
I haven't been able to find it online. Would you please
let me know how I can obtain the source code so that I
can try to build and run it?

Thank you very much for your help!

Christian Collberg
ccollberg@gmail.com
```

Figure 3: First email.

```
From: Christian Collberg <ccollberg@gmail.com>
To: first-or-corresponding-author
Cc: remaining-authors
Subject: RE: Your conference-name paper

Hi!

Sorry for bothering you again!

Can you please let me know if this implementation is available,
and from where I can obtain the code?

Thanks!
```

Figure 4: Reminder email.

of the analysis and reviews. To further facilitate data entry and reduce the risk of erroneous data, a special data entry editor was developed, shown in Figure 2.

3.1 Email Requests

If no link to source code was found online or in the article itself, an email was sent to the authors to request access to the code. The emails had the structure shown in Figure 3. In order to avoid messages being flagged as spam or making the recipient suspicious as to the motive behind the

emails, papers were removed from cosideration so that the resulting set had no overlapping author lists. If no response was received within a few weeks, a reminder email was sent, shown in Figure 4.

3.2 Build Process

Each paper for which source code was found was assigned to one of the student research assistants. The code was usually tested under the newest Ubuntu environment, unless specified otherwise by instructions accompanying the source code. Java-based code was tested under Windows 7. Each build was given 30 minutes of programmer time (excluding download and install time) to complete. If, at the end of the 30 minutes, unresolved make or compilation errors remained, the system was marked as "did not compile." If the system compiled but did not run, it was classified as "compiles but does not run." Finally, if a build executed without errors, it was marked as "runs." No attempt was made to verify whether the system produced results consistent with the claims made in the original paper.

3.3 Outliers

Given that the papers selected for analysis cover a wide variety of topics and circumstances, some papers did not fit nicely into the build process outlined above and required additional attention.

For example, one paper provided an entirely online implementation that executes with some errors. We marked that paper as "compiles but does not run." Another paper provided an API as a tool, and requires C++ to call and use. This was deemed to be out of the scope of the research. Giving the authors the benefit of the doubt, the paper was marked as "compiles and runs." There were other papers that were out of scope as well. For example, one tool required installing a custom-provided operating system on an Android device to run, and another required downloading and modifying images for Android systems to run. We gave such papers the benefit of the doubt also.

Although often unclear, we assumed that the code we found was the version that produced the results in the paper.

A small number of systemshad online implementations and thus did not need to be compiled. We marked these as "builds."

Overall we found these kind of occurrences to be rare and as such did not significantly impact the results of the study.

3.4 Detailed Workflow

Three undergraduate and one graduate research assistant were tasked with downloading and scanning papers for links to code. For each paper the information in Figure 5 was collected. The research assistants went through the following workflow:

1. Get pdf and basic information:
   (a) Look for grant support. Search for the word "grant", usually at beginning or end of the paper, possibly in the footnote or acknowledgement sections.
   (b) Extract the email addresses of authors.
   (c) Determine if any authors come from a commercial entity. Look at authors' email address and grant/financial support.
   (d) Add a link to article.

```
ARTICLE:ANALYSIS_BY[name]
ARTICLE:ANALYSIS_TIME[minutes]
ARTICLE:COMMENT[string]
ARTICLE:COMMERCIAL_EFFORT[none,part,full]
ARTICLE:GRANT_SUPPORT[none or string]
ARTICLE:NSF-SUPPORT[none or number]
ARTICLE:IMPLEMENTATION_EXISTS[unknown,hardware,yes,no]
ARTICLE:LINK[url]
ARTICLE:STATUS[not_finished,finished]
AUTHOR:EMAIL_REAL[list of strings]
BIBTEX:LABEL[string] HackettG12
AUTHOR:NAMES[list of first_last]
BIBTEX:LINK[url]
BUILD:ANALYSIS_BY[name]
BUILD:ANALYSIS_TIME[minutes]
BUILD:COMMENT[string]
BUILD:STATUS[one of {unknown,needed,not_needed,started,finished} and list of {downloaded,compiles,runs}]
EMAIL:STATUS[unknown,not_needed,not_found or list of {needed,request_1,response_1,sent_thank_you}]
PI:COMMENT_CC[string]
PI:COMMENT_TP[string]
TOOL:NAME[string]
TOOL:ARTICLE_LINK[unknown,none,url,broken and url]
TOOL:GOOGLE_LINK[unknown,none,url,broken and url]
TOOL:EMAIL_LINK[unknown,none,sent_no_url,url,broken and url]
TOOL:DATA_LINK[unknown,none,url,broken and url]
VERIFY:ANALYSIS_BY[name]
VERIFY:STATUS[unknown,needed,not_needed,started,finished]
VERIFY:COMMENT[string]
EMAIL1:CODE_AVAILABLE[yes,no,no_response]
EMAIL1:REMARK[comment]
```

Figure 5: Textual form for the data collected for a paper.

---

2. Look through Abstract/Introduction:
   (a) Skim the text to get an idea of what the paper is about.
   (b) Find out what a possible tool might be called.
   (c) Determine if the paper is backed by an implementation (code).
   (d) Search the introduction for a link to code.
3. Search for “git,” “code.google,” “http,” and “ftp” for links to code:
   (a) If you find a link in the citations search back for the citation number to see what it is.
4. Skim section headings and text for links to code:
   (a) Sections named “Implementation” or “Experiments” could have information on the tool.
5. Do a web search:
   (a) Search the name of the tool.
   (b) Search with relevant keywords.
   (c) Search each authors’ homepage and check if they link to the software.
   (d) Try to figure out what lab/research group the project belongs to and check the lab’s website.
   (e) If a slide deck related to the paper is found look for links in them (probably at end).
6. Send initial emails:

Table 1: Summary of the results of the study.

Group	#	Practical	Emails	Email Reply			Build		Run		Reproduci-
			not sent	None	"yes"	"no"	fails	OK	fails	OK	bility
ASPLOS'12	37	30	7	1	4	15	3	4	0	4	17.4%
CCS'12	76	56	18	2	15	11	12	13	4	9	23.7%
OOPSLA'12	81	66	5	1	9	15	22	23	2	21	34.4%
OSDI'12	24	24	7	0	2	7	4	6	1	5	29.4%
PLDI'12	48	48	7	1	10	14	17	9	5	4	9.8%
SIGMOD'12	46	45	20	1	8	7	6	11	2	9	36.0%
SOSP'11	28	26	7	1	3	9	7	2	0	2	10.5%
TACO'9	60	39	2	5	5	17	8	7	0	7	18.9%
TISSEC'15	13	6	0	0	1	2	2	2	0	2	33.3%
TOCS'30	14	13	0	2	0	6	3	2	0	2	15.4%
TODS'37	29	17	0	5	3	3	3	6	0	6	35.3%
TOPLAS'34	16	11	2	0	1	0	5	4	0	4	44.4%
VLDB'12	141	134	30	11	20	43	16	34	7	27	26.0%
NSF	255	227	54	12	37	58	50	53	9	44	25.4%
No NSF	358	288	51	18	44	91	58	70	12	58	24.5%
Academic	415	350	63	20	62	86	85	96	12	84	29.3%
Joint	149	130	36	7	17	45	19	23	8	15	16.0%
Industrial	49	35	6	3	2	18	4	4	1	3	10.3%
Conferences	481	429	101	18	71	121	87	102	21	81	24.7%
Journals	132	86	4	12	10	28	21	21	0	21	25.6%
Total	613	515	105	30	81	149	108	123	21	102	24.9%

   (a) An initial email needs to be sent if: the paper is backed by code, and no link was found.
   (b) Exclude papers so that no more than one email is sent to any author.
   (c) Send a reminder email if a response to the first one is not received.

7. Build code:
   (a) Follow link and download code.
   (b) Attempt to build. Install any missing libraries. Fix any broken search paths.
   (c) Attempt to run.
   (d) Abort after 30 minutes.

8. Notes:
   (a) If a link was found through a web search go back and check the paper again to make sure it was not there.
   (b) It can be complicated to determine when there is a larger project of which the current paper is a subset. In that case the paper may refer to the larger project as though it were a separate subject when in fact their current code is included with it.

4 Results

Table 1 and Figures 6, 7, 8, and 9 show our results.

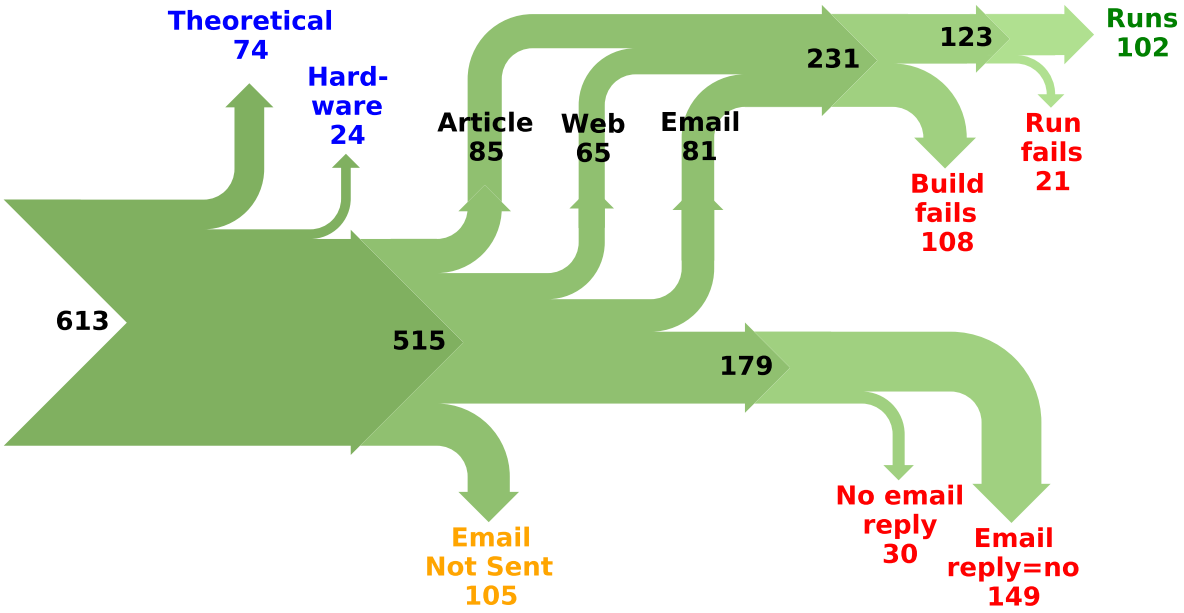


Figure 6: Study result. Blue numbers represent papers that were excluded from consideration, green numbers papers that are reproducible, red numbers papers that are non-reproducible, and orange numbers represent papers for which we could not conclusively determine reproducibility (due to our restriction of sending at most one email to each author).

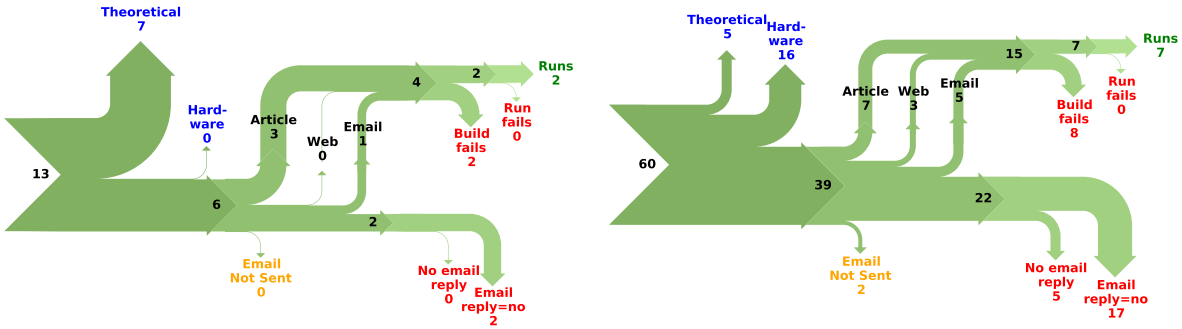
4.1 Does NSF Funding Affect Sharing?

The National Science Foundation's (NSF) Grant Policy Manual[6] states that

> **b.** Investigators are expected to share with other researchers, at no more than incremental cost and within a reasonable time, the primary data, samples, physical collections and other supporting materials created or gathered in the course of work under NSF grants. Grantees are expected to encourage and facilitate such sharing. [...]
>
> **c.** Investigators and grantees are encouraged to share software and inventions created under the grant or otherwise make them or their products widely available and usable.
>
> **d.** NSF normally allows grantees to retain principal legal rights to intellectual property developed under NSF grants to provide incentives for development and dissemination of inventions, software and publications that can enhance their usefulness, accessibility and upkeep. Such incentives do not, however, reduce the responsibility that investigators and organizations have as members of the scientific and engineering community, to make results, data and collections available to other researchers.

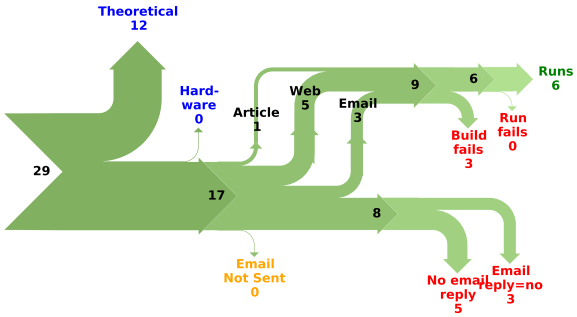
Even so, we see no difference between those authors who are supported by the NSF and those who are not, in their willingness to share.

[6] `http://www.nsf.gov/pubs/manuals/gpm05_131/`

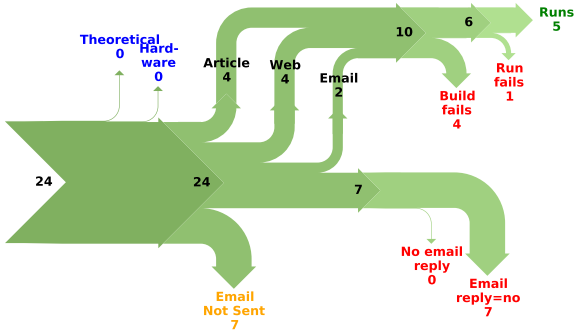


(a) TISSEC

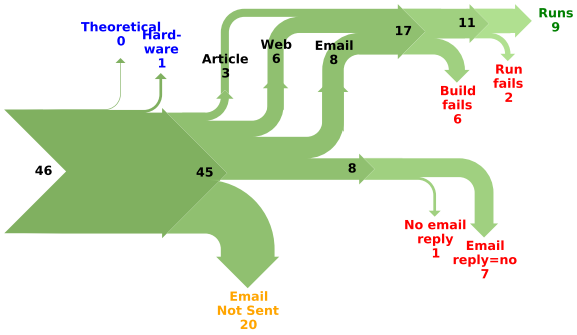
(b) TACO



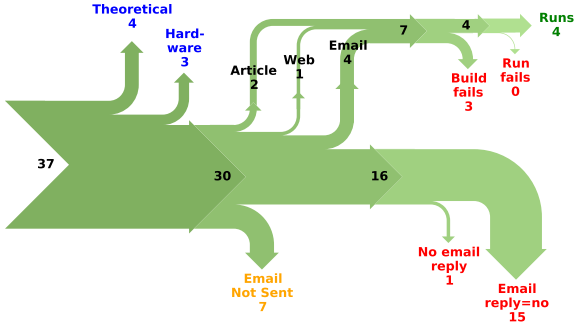
(c) TODS



(d) OSDI



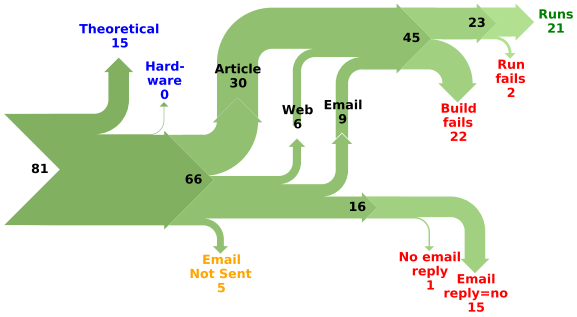
(e) SIGMOD



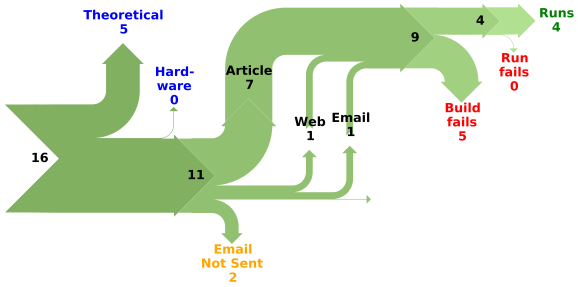
(f) ASPLOS

Figure 7: Results for individual publications.

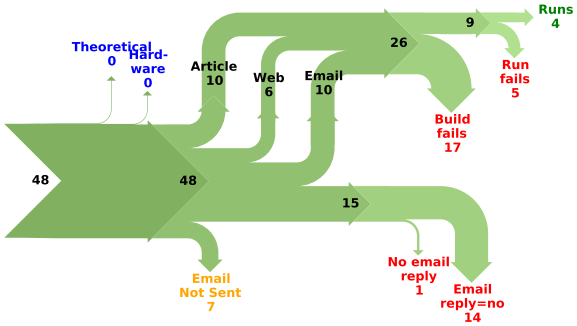
Does Industry Involvement Affect Sharing? Unsurprisingly, papers with only authors from industry have a low rate of reproducibility,, and papers with only authors from academia have a



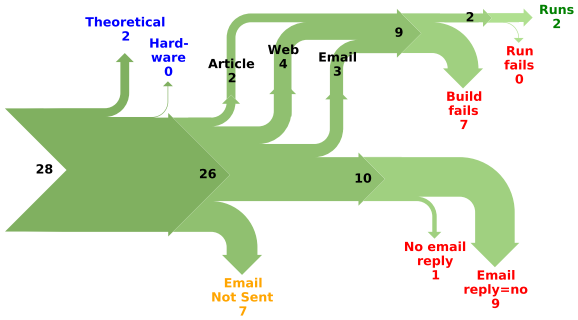
(g) OOPSLA



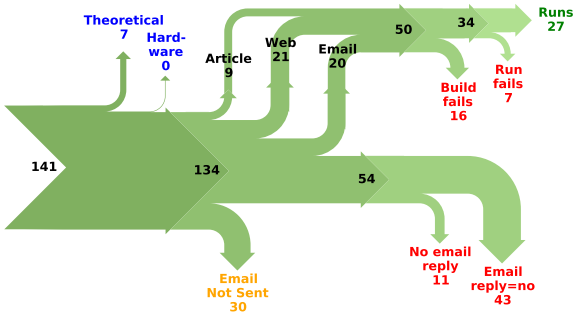
(h) TOPLAS



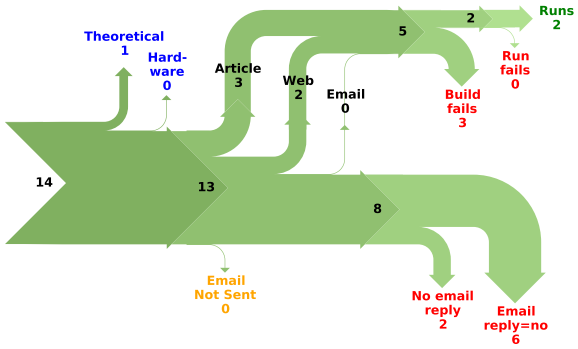
(i) PLDI



(j) SOSP



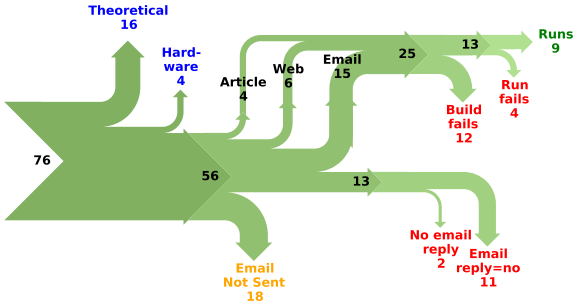
(k) VLDB



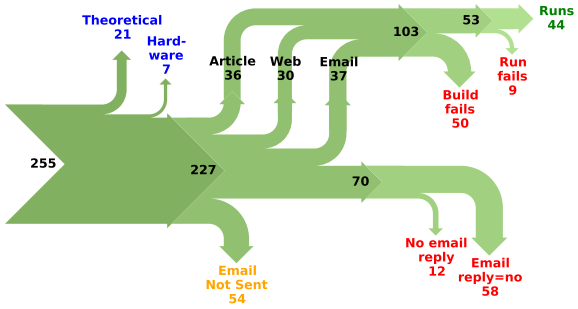
(l) TOCS

Figure 8: Results for individual publications.

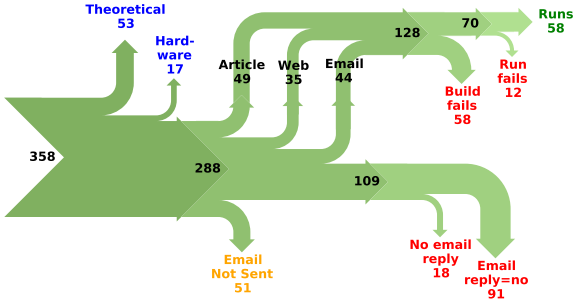
higher than average rate. The reasons why joint papers also have a very low rate is not immediately obvious: the industrial partner might have imposed intellectual property restrictions, the research



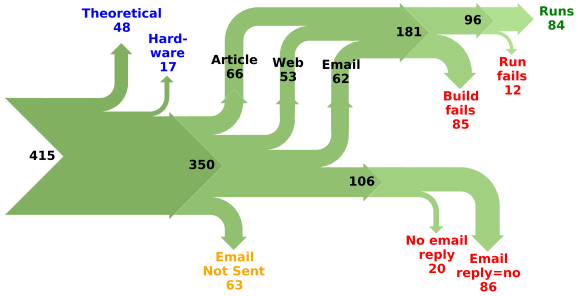
(m) CCS



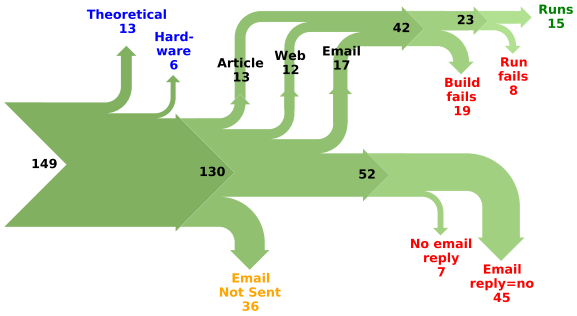
(n) NSF



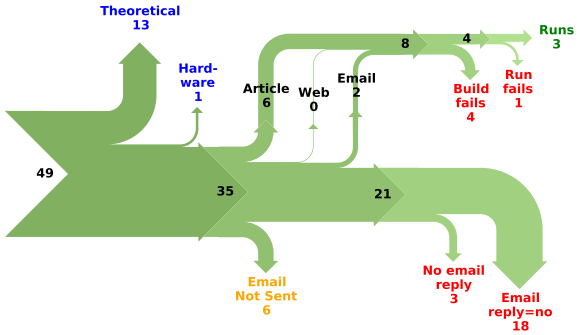
(o) No NSF



(p) Academic



(q) Joint



(r) Industrial

Figure 9: Results for individual publications.

could be the result of a student's summer internship, etc.

Table 2: Build error summary.

error	count	percentage
distribution is missing files	11	18%
prerequisite failed to build	3	4.9%
unavailable environment	3	4.9%
other errors	17	27.9%
incomplete documentation	9	14.8%
missing third party package	7	11.5%
build:error comment[not needed,comment]	2	3.3%
runtime error	9	14.8%
Total	61	

internal compiler error: Compilation terminated with the message "Internal Compiler Error."

unavailable environment: The build failed due to an unavailable version of a particular piece of software, compiler etc.

missing third party package: A particular piece of software, package or tool required to run the build was not found.

distribution file is missing: The author did not make available files that should be part of the build, such as a missing header file or a missing file with input data to run the code.

incomplete documentation: Documentation necessary to build the software is incomplete or missing.

runtime error: The code compiles and fails to run, due, for example, to a segmentation fault or a runtime exception.

prerequisite failed to build: A pre-requisite for a tool could not be successfully built by the team in under 30 minutes.

other errors: The error could not be categorized.

4.2 But, Does it Build?

It is not unreasonable to believe that build numbers could be dependent on the skills of our team of undergraduate and graduate student research assistants. It may well be that, given more time and better background, more systems would build successfully. In our experiments we instructed the students to spend no more than 30 minutes on building the systems. In many cases this involved installing additional libraries and compilers, editing makefiles, etc. The students were also instructed to be liberal in their evaluations, and, if in doubt, mark systems as buildable.

The types of build errors encountered can be found in Table 2.

4.3 So, What Were Their Excuses? (Or, The Dog Ate My Program)

The email responses we received were generally pleasant, appologetic if code could not be provided, and accomodating. Below are a few representative examples of emails from authors who turned down our request.

4.3.1 Versioning Problems

Version problems appear to be a major obstacle to reproducible research. In most cases we could not determine if the code we downloaded corresponded to the version in the paper, and often, neither could the author:

> Thanks for your interest in the implementation of our paper. The good news is that I was able to find some code. I am just hoping that it is a stable working version of the code, and matches the implementation we finally used for the paper. Unfortunately, I have lost some data when my laptop was stolen last year. The bad news is that the code is not commented and/or clean. So, I cannot really guarantee that you will enjoy playing with it.

> Attached is the ⟨system⟩ source code of our algorithm. I'm not very sure whether it is the final version of the code used in our paper, but it should be at least 99% close. Hope it will help.

4.3.2 Code Will be Available Soon

Many responses we received indicated that the code corresponding to the published paper was too bad to be released, that new code was being worked on, and once done, we could get access to the cleaned up system:

> Thank you for your interest in our work. Unfortunately the current system is not mature enough at the moment, so it's not yet publicly available. We are actively working on a number of extensions and things are somewhat volatile. However, once things stabilize we plan to release it to outside users. At that point, we would be happy to send you a copy.

While the authors should be commended for eventually making their code available, it should be pointed out that for *reproducibility*, such delayed releases are unacceptable: it will never be possible for a reviewer or reader of the paper to verify the results that are actually presented in the paper.

4.3.3 No Intention to Release

A few authors acknowledged that they never had the intention to make the code available:

> I am afraid that the source code was never released. The code was never intended to be released so is not in any shape for general use.

4.3.4 Programmer Left

In some cases, the one person who understood the system left:

> ⟨STUDENT⟩ was a graduate student in our program but he left a while back so I am responding instead. For the paper we used a prototype that included many moving pieces that only ⟨STUDENT⟩ knew how to operate and we did not have the time to integrate them in a ready-to-share implementation before he left. Still, I hope you can build on the ideas/technique of the paper. Regards,

4.3.5 Bad Backup Practices

In one case, the system physically ceased to exist:

> Thanks for your interests in this paper. Unfortunately, the server in which my implementation was stored had a disk crash in April and three disks crashed simultaneously. While the help desk made significant effort to save the data, my entire implementation for this paper was not found. I do have reports generated by the tool for ⟨SYSTEM⟩. If you are interested in these reports, I can send them to you. Sorry for that.

4.3.6 Commercial Code

Research done by employees of commercial entities were often not able to release their code

> Since this work has been done at ⟨COMPANY⟩ we don't open-source code unless there is a compelling business reason to do so. So unfortunately I don't think we'll be able to share it with you.

but sometimes the author added a helpful suggestion:

> The code owned by ⟨COMPANY 1⟩, and AFAIK the code is not open-source. Your best bet is to reimplement :(Sorry.

4.3.7 Proprietary Academic Code

Even university researchers at universites had licensing issues

> Thank you for the interest in our ⟨CONFERENCE⟩ contribution.
> Unfortunately, the ⟨SYSTEM⟩ sources are not meant to be opensource
> (the code is partially property of ⟨UNIVERSITY 1⟩,
> ⟨UNIVERSITY 2⟩ and ⟨UNIVERSITY 3⟩.)
>
> In particular, while there is interest, I do not have the liberty of
> making available the source code at my current institution
> (⟨UNIVERSITY 4⟩).
>
> If this will change I will let you know, albeit I do not think there
> is an intention to make the ⟨SYSTEM⟩ sources opensource in the
> near future.

or put restrictions on the release of the code:

> Thanks for your interest.
>
> We haven't yet open-sourced the software, but we are making a collaboration
> release available to academic partners.
>
> If you're interested in obtaining the code, we only ask for a description of
> the research project that the code will be used in (which may lead to some
> joint research), and we also have a software license agreement that the
> University would need to sign.

4.3.8 Unavailable Subsystems

Some systems were built on top of other systems which were either not publically available

> We implemented and tested our sparse analysis technique on top of a
> commercialized static analysis tool. So, the current implementation
> is not open to public. Sorry for this.

or obsolete:

> Currently, we have no plans to make the scheduler's source code
> publicly available. This is mainly because ⟨ANCIENT OS⟩ as such
> does not exist anymore, and as a result installing this OS on new
> hardware is a bit troublesome; not to mention how tedious it is to
> prepare a new machine for OS development under a discontinued version
> of ⟨ANCIENT OS⟩ such as the one we used (official package
> repositories no longer exist). Although some researchers asked us to
> disclose the source code, we strongly believe that the efforts to make
> the code available and fully documented now are not worth it since few
> people would manage it to get it to work on new hardware.

4.3.9 Multiple Reasons

Sometimes, there were multiple reasons why not to release the code, including licensing, bad code, and reliance on other systems:

> Sorry we haven't released our code because it is too hard to maintain. A large part of our code contains/relies on the code from other tools (probably old versions), such as ⟨LONG LIST OF SYSTEMS⟩. It would require huge amount of effort to make our code work with the latest versions of these tools. We are not using the code at ⟨COMPANY 1⟩ or ⟨COMPANY 2⟩ so we didn't spend time on maintaining it. Besides, we cannot release our code with the tools in one package since they are protected by licenses.

4.3.10 Intellectual Property

Some authors were worried about how their code might be used:

> ... I hope the above will be proved of help. If these suggestions are not helpful in your efforts, we could provide our implementation but please kindly note that:
> - We won't be able to provide further support in a timely manner. Of course, installation instructions and initialization parameters will accompany the implementation
> - We would like to be notified in case the provided implementation will be utilized to perform (and possibly publish) comparisons with other developed techniques. If this is indeed the case, please note that we would prefer to look into the extracted results and be able to comment on the comparison results before their publication.
>
> The reason for the second provision is that, based on earlier (bad) experience, we would like to make sure that our implementation is not used in situations that it was not meant for. For instance, in this particular case, our code has not been optimized for speed, so it makes no sense to use it for performance measurements over streaming data. In other words, we would like to have some say on how our code is used, especially if the results are to be used in a paper submission.

4.3.11 Research vs. Sharing

Producing artifacts which are solid enough to be shared is clearly labor intensive, and one researcher explained how he had to make a draconian choice:

> Thank you for your email to ⟨STUDENT⟩, expressing interest in our work.
> Since giving you ⟨system⟩ would necessitate us also giving you our complex
> Simulation infrastructure, which has been developed over a lot of
> years by a large number of my former and current PhD students, he knew
> that I would want it forwarded it to me for answering.
>
> Our simulator is very complex, and continues to become more complex
> as more PhD students add more pieces to it. The result is that without
> a lot of hand holding by senior PhD students, even our own junior PhD
> students would find it unusable. In the past when we attempted to share
> it, we found ourselves spending more time getting outsiders up to speed
> than on our own research. So I finally had to establish the policy that
> we will not provide the source code outside the group.
>
> On the other hand, ⟨STUDENT⟩ would be very happy to answer any questions you
> have about ⟨SYSTEM⟩ if you decide to implement it on your own simulation
> infrastructure. In fact, both ⟨STUDENT⟩ and I would welcome that opportunity.
> For us, in fact, it would show that the ideas in ⟨SYSTEM⟩ can be validated on
> another Simulator developed by an outside group. Feel free to take ⟨STUDENT⟩
> up on this offer if you wish.
>
> I wish you success in your undertakings and hope that ⟨STUDENT⟩'s help
> understanding ⟨SYSTEM⟩ will be a win-win for us both.

4.3.12 Security and Privacy

Unlike those working in other fields, Computer Security researchers have to contend with the possible negative consequences of making their code public:

> Thanks for your interest. However, we have an agreement with the utility company,
> and we cannot release the code because of the potential privacy risks to the general
> public.

4.3.13 Poor Design

One author admitted to poor initial design:

> The code used to implement the ⟨CONFERENCE⟩ paper is
> complete, but hardly usable by anyone other than the authors. This is
> due in large part due to our decision to use Template Haskell for the
> input language. The error messages which are produced by the compiler
> are useless to anyone not fluent in both Haskell, BSV, and the
> compiler architecture.

4.3.14 Too Busy to Help

Obviously, some people were too busy to help out understanding the code:

$$
\begin{array}{rcll}
\rho & \in & \{\texttt{code}, \texttt{data}, \texttt{media}, \texttt{docs}, \ldots\} & \textit{(resources)} \\
\alpha & \in & \{\texttt{no access}, \texttt{access}\} & \textit{(availability)} \\
\phi & \in & \{\texttt{source}, \texttt{binary}, \texttt{service}\} & \textit{(distribution form)} \\
\xi & \in & \{\texttt{free}, \texttt{non-free}\} & \textit{(expense)} \\
\tau & \in & \textit{ISO 8601 Date} & \textit{(expiration date)} \\
\lambda & \in & \textit{RFC 3986 URI} & \textit{(uri)} \\
\pi & \in & \{\texttt{L1}, \texttt{L2}, \texttt{L3}\} & \textit{(support level)} \\
\sigma & \in & \textit{string} & \textit{(license-string)} \\
\textit{res} & ::= & \rho(, \rho)^* : \alpha[, \xi][, \phi][, \tau][, \sigma] \ ; & \textit{(sharing)} \\
\textit{loc} & ::= & \lambda \ ; & \textit{(location)} \\
\textit{sup} & ::= & \texttt{support} : \pi, \xi[, \tau] \ ; & \textit{(technical support)} \\
s & ::= & \textbf{Sharing } \textit{loc}^+ \ \textit{res}^* \ \textit{sup}^* & \textit{(contract)}
\end{array}
$$

Figure 10: Grammar generating sharing specifications.

```
Most importantly, I do not have the bandwidth to help anyone come up to
speed on this stuff.
```

5 Recommendations

As we have seen, there are many ways to improve the state of reproducibility in Computer Systems research. We could simply, as some conferences have done, require that along with every paper submitted for publication the authors should attach the corresponding code, perhaps in the form of a virtual machine image. Or, we could build special tools that help authors produce code that can run reliably and with reproducible results, regardless of the execution environment. Or, we could build web sites that allow authors to make their code available to colleagues in perpetuity.

Unfortunately, based on the study presented in this paper, it is our belief that it is currently not realistic to expect Computer Systems researchers always to make their code available to others for scrutiny. The reasons for the reluctance to share code are manyfold: the code may not be clean enough for public distribution; the authors may be considering commercialization of their work and would prefer to keep the code out of the hands of competitors; (part of) the code may not be publishable because of licensing restrictions; the authors may subscribe to the "publish and forget" style of scientific research; or, the code may be so severely broken that making it available to reviewers would lower the chances of the paper being accepted for publication. We may question the scientific ethics of those who espouse these types of rationalizations but they are, nevertheless, not uncommon in the community.

Our proposal is therefore much more modest. Rather than forcing authors to make their systems available, we propose instead that *every article be required to specify the level of reproducibility a reader or reviewer should expect*. We will term such a specification **sharing**. The sharing specification should be provided in the article header (along with keywords and classification terms) *at the time of submission* (allowing reviewers to take reproducibility into account when deciding on whether to recommend acceptance or rejection) as well as in the camera ready version (allowing readers to know if, and how, to access the relevant code, data, and other resources). The specification can be construed as a *contract* between an author and their reader/reviewer in which (a)

Keywords
Mobile, Ubiquitous Computing, Sensors

ACM Classification Keywords
H5.m. Information interfaces and presentation:
Miscellaneous

General Terms
Design Document, Work–in–Progress

Sharing
mailto:alice@example.com; http://mobile-project.example.com;
code:access,free,source,Apache 2.0;
support:L1,free,perpetuity; support:L2,non–free,2014–12–31;

Figure 11: Example of paper heading. The sharing section specifies that the paper is backed by code but no other resources, that the code will be provided for free in source form, that support will be provided until the end of 2014, and that the code is distributed under the Apache license.

---

the author commits to making available certain resources that were used in the research leading up to the paper and (b) the reader/reviewer commits to take these resources into account when evaluating the contributions made in the paper.

5.1 Syntax of Sharing Specifications

In Figure 10 we give a grammar for one possible sharing specification. Figure 11 gives an example of a paper header with a sharing specification generated from this grammar. A specification consists of three parts: the external resources that back up the results in the paper, locations where these resources can be found or ways to contact the authors, and the level of technical support the authors will provide.

Resources can include code, data, media, etc. For each resource the specification states if they will be made accessible to the community, if there is a cost involved in obtaining the resources, an optional deadline after which the resources may no longer be available, an optional licensing scheme, and whether a resource is available in source or binary form or is accessed as a service. Here, "service" represent any dissemination method that gives a user indirect rather than direct access to the resource. Examples include programs executing as a web service, media streamed from a server, or authors guaranteeing that they will run the code by hand, on input data provided by the user, and then provide them with the resulting output. An absence of an expiration date indicates the resource will be available in perpetuity.

The `support` field specifies the level of technical support the authors will provide, for how long, and whether support is free or not. There are three levels of support: `L1` indicates the authors will resolve issues with downloading, installing, and executing the program; `L2` indicates the authors will maintain the program, fixing bugs, and upgrade as necessary as language specifications and operating systems evolve; `L3` indicates a willingness on part of the authors to port the program to new environments, improve algorithms and performance, add new features, etc. Multiple support levels can be specified. An absense of the `support` field indicates no support is offered. An absence of an expiration date indicates that support will be available in perpetuity.

For each resource the specification can specify the license under which a resource is provided. An

absense of the license field indicates that the resource is provided under an "anything goes" license. It should be noted that *sharing*, as defined in this paper, is different from *licensing*. The sharing specification in Figure 10 represents a commitment an behalf of the author of a paper to make resources available to the wider community for scrutiny. A *license*, on the other hand, describes the actions allowed on these resources, such as modification, re-distribution, reverse engineering, etc. Under some circumstances it is necessary to specify both licensing and sharing. For example, if code is distributed in binary form only and the license prohibits reverse engineering, the community's ability to verify that the actions performed by the software is consistent with what is described in the publication is diminished. Similarly, reproducibility is hampered by code that makes use of libraries whose license prohibits re-distribution.

5.2 Example Sharing Specifications

The sharing specification in Figure 10 is designed to handle most common situations, but obviously represents ta tradeoff between expressiveness and conciseness. Since our hope is that in the future every paper will contain such a specification, we do not want it to take up too much space nor do we want writing it to be too onerous. In most cases, authors will find that a very terse specification will suffice. For a purely theoretical paper, i.e. one not backed by any external resources, the specification would be simply

```
Sharing
    bob@cs.example.edu
```

i.e. it defaults to providing "corresponding author" information. In the common situation where a paper is backed by code and data, the authors expect to put them both up on their web site, never taking them down, not providing any support, and not worrying about licensing, the specification would be

```
Sharing
    http://project.example.edu; code, data:access, free, source;
```

In a situation where the authors are employed by a commercial organization and, as a result, the code is proprietary and cannot be shared, the benchmarks used are freely shared, and limited support in how to download, install, and run the benchmarks will be given until the beginning of 2015, the specification would be

```
Sharing
    http://project.example.com; code: no access;
    data:access, free, source; support:L1, free, 2015-01-01;
```

There may certainly be complex situations that cannot be expressed by the grammar in Figure 10, but these can be handled by the authors providing additional information in a footnote.

5.3 Lessons Learned

From the work that has lead up to this paper, in particular the responses to our email queries, we have learned the following valuable lessons:

1. **Unless you have compelling reasons not to, plan to release the code.** It is the right thing to do, and if you start with this mind-set from the beginning of the project, the amount of extra work will likely be negligible.

2. **Students will leave, plan for it.** When building the system keep in mind that the code should outlive both you and the student.

3. **Create permanent email addresses.** You and your students will most likely be changing jobs a few times during your career. While some schools will keep old email addresses around, or forward email, you cannot count on it. Create email addresses that you know will be permanent throughout your working life and use them in all professional correspondence.

4. **Create project websites.** These are more likely to remain functional over time than email addresses. Put the URL in the paper. Be prepared to upload code and test data to your web sites at the same time as you upload the paper describing your system to a conference site for review.

5. **Use a source code control system.** Whenever you submit or publish a paper, set a label on the corresponding code version so that you can easily recreate it.

6. **Backup your code.**

7. **Resolve licensing issues.** If you anticipate problems start the licensing process early, so that you are able to release the code at the same time as the paper is submitted for publication.

8. **Keep your promises.** If your grant application states that you will be sharing code with the community, plan for keeping that promise.

9. **Plan for longevity.** Projects may live on for a long time, with many students building on the code. Plan for this at the onset of the project, by setting up the appropriate directory structures, plug-in architectures, etc., which will allow the project to grow.

10. **Avoid cool but unusual designs.** Unless you have a compelling reason to do otherwise, stay with standard operating systems, programming languages, and tool chains.

11. **Plan for Reproducible Releases.** Use the same techniques that Release Managers in industry use to ensure consistent and reproducible builds. For example, check in your entire tool chain (compilers, linkers, libraries, etc.) into your source code control system, and start any release by building the tool chain from scratch. There is an extra startup cost when beginning a new project, but this will be paid off over time. It will, for example, make it easier for new students to join the project.

6 Conclusions

We believe that, if generally adopted, the sharing specification proposed in this paper will have positive effect on researchers' willingness to share their code and data. First of all, knowing that reviewers may take a dim view of a paper whose sharing specification is `code:no access; data:no access`, may serve as an incentive for authors to decide, at the onset of a project, to ensure that the computational resources produced will be solid enough to be shared. Secondly, a fellow researcher can point to the sharing specification—which, essentially is a contract between authors and readers—and demand that this contract be fulfilled, and code be shared.

References

[1] W. M. 0001, I. Rojas, A. Eberhart, P. Haase, and M. Schmidt. A-r-e: The author-review-execute environment. *Procedia CS*, 4:627–636, 2011.

[2] J. Austin, T. Jackson, M. Fletcher, M. Jessop, B. Liang, M. Weeks, L. Smith, C. Ingram, and P. Watson. Carmen: Code analysis, repository and modeling for e-neuroscience. *Procedia CS*, 4:768–777, 2011.

[3] G. R. Brammer, R. W. Crosby, S. Matthews, and T. L. Williams. Paper mch: Creating dynamic reproducible science. *Procedia CS*, 4:658–667, 2011.

[4] C. S. Collberg, A. Huntwork, E. Carter, and G. M. Townsend. Graph theoretic software watermarks: Implementation, analysis, and attacks. In J. J. Fridrich, editor, *Information Hiding*, volume 3200 of *Lecture Notes in Computer Science*, pages 192–207. Springer, 2004.

[5] S. Fomel and J. F. Claerbout. Guest editors' introduction: Reproducible research. *Computing in Science and Engineering*, 11(1):5–7, 2009.

[6] M. Gavish and D. Donoho. A universal identifier for computational results. *Procedia CS*, 4:637–647, 2011.

[7] R. Gentleman. Reproducible research: A bioinformatics case study. *Statistical Applications in Genetics and Molecular Biology*, 4(1):1034, 2005.

[8] P. V. Gorp and S. Mazanek. Share: a web portal for creating and sharing executable research papers. *Procedia CS*, 4:589–597, 2011.

[9] D. Koop, E. Santos, P. Mates, H. T. Vo, P. Bonnet, B. Bauer, B. Surer, M. Troyer, D. N. Williams, J. E. Tohline, J. Freire, and C. T. Silva. A provenance-based infrastructure to support the life cycle of executable papers. *Procedia CS*, 4:648–657, 2011.

[10] J. Kovačević. How to encourage and publish reproducible research. volume IV, pages 1273–1276, Apr. 2007.

[11] S. Li-Thiao-T. Literate program execution for reproducible research and executable papers. *Procedia CS*, 9:439–448, 2012.

[12] N. Limare. Running a reproducible research journal, with source code inside. In *ICERM Workshop*, 2012.

[13] Y. L. S. R. on Data and C. Sharing. Reproducible research. *Computing in Science and Engineering*, 12(5):8–13, 2010.

[14] M. Schwab, N. Karrenbach, and J. Claerbout. Making scientific computations reproducible. *Computing in Science Engineering*, 2(6):61–67, 2000.

[15] V. Stodden. Enabling Reproducible Research: Open Licensing for Scientific Innovation. *Social Science Research Network Working Paper Series*, Mar. 2009.

[16] V. Stodden. The legal framework for reproducible scientific research: Licensing and copyright. *Computing in Science and Engineering*, 11(1):35–40, 2009.

[17] V. Stodden. The Scientific Method in Practice: Reproducibility in the Computational Sciences. Technical Report Working Paper 4773-10, MIT Sloan School of Management, Feb. 2010.

[18] V. Stodden, C. Hurlin, and C. Perignon. Runmycode.org: A novel dissemination and collaboration platform for executing published computational results. In *eScience*, pages 1–8. IEEE Computer Society, 2012.

[19] V. Stodden, I. Mitchell, and R. LeVeque. Reproducible research for scientific computing: Tools and strategies for changing the culture. *Computing in Science and Engineering*, 14(4):13–17, 2012.

[20] P. Vandewalle, J. Kovacevic, and M. Vetterli. Reproducible research in signal processing - what, why, and how. *IEEE Signal Processing Magazine*, 26(3):37–47, May 2009.

A Anecdotes

A.1 Anecdote 1

In 2001 the first author (Collberg) asked some colleagues at a major corporate research lab for access to a system which, eventually, would be published in a highly-regarded workshop:

```
... Do you have software available for us to play with?
```

The response was negative:

```
As you may guess, implementations of our papers (even the one on
⟨technique⟩) may not be made available. At least the
⟨technique⟩ has a chance to be accessible via applications that
may use it in whatever form.

Sorry about that.
```

As a result, our students spent a year re-implementing their system, making "reasonable" inferences in the many cases when the paper was unclear, in order to repudiate the purported results. Our conclusions were eventually published [4]. However, because of the many implementation details that were glossed over in the original paper, it is highly likely that we did not faithfully reproduce the original system, and this diminishes the value of our results.

A.2 Anecdote 2

During 2012 the first author (Collberg) together with colleagues submitted a paper to a series of security conferences: *IEEE Security and Privacy*, *CCS*, and *USENIX Security*. In the paper we showed a new attack against a defense that had been published in a top-tier security conference in 2009.[7] The reviewers of all three conferences pointed to a number of problems with the paper and it was ultimately rejected. One problem, in particular, was a less-than-conclusive security evaluation:

[7] In the following, we use ⟨CONFERENCE⟩ as the name of the conference in which this paper was published, ⟨PAPER⟩ as the title of the paper, ⟨SYSTEM-R⟩ as the name of the system described in the paper, and ⟨MAIN STUDENT⟩, ⟨SECOND STUDENT⟩, ⟨JUNIOR-PROFESSOR⟩, ⟨PROFESSOR⟩ as the authors. ⟨UNIVERSITY⟩ is the name of the public university at which they were studying/employed, at a top-10 Computer Science Department. ⟨COMPANY⟩ is the name of the company where ⟨MAIN STUDENT⟩ is now employed. ⟨SYSTEM-D⟩ is the name of the system we built and tried to publish.

```
It is unfortunate that ⟨MAIN STUDENT⟩ et al. did not make
their tool publicly available, and it is great that the authors in
this paper promise to make their tools available.

The evaluation has a "built here" flavor--both the implementation and
analysis parts were done by the authors without relying on
experimental standards or using other implementations or benchmarks in
a significant way. This to me is a major threat to validity.

Interesting area to explore and useful tools to build, but overall
somewhat less than conclusive, the deployment is not seriously
considered, and the evaluation is of an "internal" kind, lacking
external benchmarks or tools that are compared directly.

2. The security evaluation does not sufficiently justify/support the
claim (in the abstract) that their new tool is resistant to known
attacks against obfuscation techniques
```

The problem was that we did not have access to ⟨SYSTEM-R⟩ and were not able to pitch it against our own tool, ⟨SYSTEM-D⟩, in order to prove, conclusively, that we would overcome the techniques published in ⟨PAPER⟩. We thus had to resort to a hand-wavy security argument which the reviewers did not find convincing.

A.2.1 First Emails

On October 14, 2012 we had sent this email to the authors:

```
Hi!

I'm Christian Collberg from the University of Arizona. I was
wondering if you have a copy of ⟨SYSTEM-R⟩ that we could
try out? We have a new ⟨SYSTEM⟩ and I'd like to see how
well your system handles our ⟨TECHNIQUES⟩.

Thank you!

Christian
```

Three of the four email addresses extracted from the author list in the paper failed, the fourth, to ⟨PROFESSOR⟩, appeared to work. We received no response.

In February 19, 2013, we tried again:

```
Hi again!

I'm Christian Collberg from the University of Arizona. I wrote
to you earlier to see if I could have a copy of ⟨SYSTEM-R⟩ so
that we could try it out, but never received a reply. We have a
new ⟨SYSTEM⟩ and I'd like to see how well your system handles
our ⟨TECHNIQUES⟩.
```

Again, we received no response from ⟨PROFESSOR⟩, and the remaining addresses still bounced, as expected.

A.2.2 Trying to Reimplement

By September we had gotten frustrated with the situation, and decided to go ahead and reimplement ⟨SYSTEM-R⟩ ourselves. We started reading the conference paper, the technical report on which it was based, and ⟨MAIN STUDENT⟩'s PhD thesis in earnest, and painstakingly tried to reproduce their implementation. We soon ran into some problems, and on September 11, 2013 we sent following email:

> Hi!
>
> I'm attempting to reimplement ⟨SYSTEM-R⟩, and as I'm working my way through
> the ⟨PAPER⟩ and tech-report, I have come across some issues that
> I hope you would be able to help me with:
>
> 1) [...]
>
> 2) [...]
>
> 3) In algorithm 2, in the ⟨PAPER⟩, in the
> Initialization step, you write ⟨FORMULA INVOLVING σ⟩
> where σ isn't defined .... In the tech report you write for
> the same step ⟨DIFFERENT FORMULA INVOLVING σ⟩.
>
> Can you enlighten me as to what is going on in this step?
>
> 4) [...]
>
> 5) In ... Section 3.1.3 you write ⟨FORMULA⟩
> ... [which] doesn't typecheck. Did you mean
> ⟨DIFFERENT FORMULA⟩
> ?
>
> 6) The function ⟨FUNCTION⟩ appears to be defined in Section 3.1
> but never used.
>
> Thank you for your help,
>
> Christian Collberg

This time we had spent significant effort tracking down working email addresses. As it turns out, both students and the junior faculty member had left for industrial research positions, and, apparently, had not had their email forwarded. We found email addresses from ⟨SECOND STUDENT⟩ and ⟨JUNIOR-PROFESSOR⟩ and on September 12, 2013, we received the following response from ⟨JUNIOR-PROFESSOR⟩:

> I'm going to redirect you to ⟨MAIN STUDENT⟩, as he was the [then-] grad student who did the actual work. We haven't stayed in touch and I don't have his current email address, but you can probably ping him over LinkedIn: ⟨URL⟩

We tried to connect with ⟨MAIN STUDENT⟩ on LinkedIn, but he would not respond.

Later the same day, we received a second clarifying email from ⟨JUNIOR-PROFESSOR⟩:

> I unfortunately have few recollections of the work. I just went and pulled up a PDF of the online tech report, and it was a bit like seeing a new paper for the first time. I wasn't able to immediately see answers to your questions, and if I sat down to reread it all again, I'd probably just end up with the same questions you asked and no new answers.
>
> ⟨MAIN STUDENT⟩'s writing style had a tendency to be verbose, so we were continually pruning. That may have impacted the final result given things like missing sigma definitions. Apologies if we accidentally deleted some important bits.
>
> Last I knew (which was several years ago), he was in ⟨RESEARCHER's⟩ group at ⟨COMPANY⟩. You might try reaching him via ⟨RESEARCHER⟩.
>
> I'd also consider him to be a member of the ⟨UNIVERSITY⟩'s PTSD[a] club. He may be deliberately ignoring communication related to ⟨UNIVERSITY⟩.

[a] PostTraumatic Stress Disorder.

On September 19, 2013, we found the main student's corporate email address by strong-arming an acquaintance also working at ⟨COMPANY⟩, who was not supposed to hand it out. On September 16, 2013, satisfied that we finally had a working email address for the main author of the paper, we sent ⟨MAIN STUDENT⟩ a query for the code and a request to clear up the confusions in the paper. We received no reply.

A.2.3 Formal Request for Source Code

We next made a formal request to ⟨PROFESSOR⟩ under the open records act of the state in which ⟨UNIVERSITY⟩ is located:

```
Dear Prof. ⟨PROFESSOR⟩,

My name is Christian Collberg, and I'm a Professor
of Computer Science at the University of Arizona.

I have on several occasions emailed you and your
research group asking for code, data, and information
related to the ⟨SYSTEM-R⟩ project. I have received no response
from you.

I am therefore making a formal request under the
   ⟨STATE⟩ OPEN RECORDS ACT
   ⟨URL⟩
for
   ALL SOURCE CODE, NOTES, AND TEST DATA
   RELATED TO THE ⟨SYSTEM-R⟩ SYSTEM
as published in
   * ⟨PAPER⟩
   * ⟨TECHNICAL REPORT⟩
   * ⟨PHD THESIS⟩
and supported under federal grants

   * ⟨MILITARY AGENCY⟩ GRANT #...
   * ⟨NSF GRANT #...⟩
   * ⟨NSF GRANT #...⟩

Note that the 2005 NSF Grant Policy Manual
   http://www.nsf.gov/pubs/manuals/gpm05_131/gpm7.jsp#734
under which your grants were funded, states that
   b) Investigators are expected to share with other researchers, at no
      more than incremental cost and within a reasonable time, the primary
      data, samples, physical collections and other supporting materials
      created or gathered in the course of work under NSF grants. Grantees
      are expected to encourage and facilitate such sharing.
   c) Investigators and grantees are encouraged to share software and
      inventions created under the grant or otherwise make them or their
      products widely available and usable.
I am willing to pay applicable fees. These records are sought in
furtherance of scholarly research, and I am employed by an Educational,
Non-commercial Scientific Institution. Therefore, I ask that fees,
other than duplication fees, be waived.

Thank you,
Prof. Christian Collberg
⟨ADDRESS, EMAIL, PHONE⟩
CC: ⟨DEPARTMENT CHAIR⟩
```

No response was forthcoming within the 3 day limit imposed by the statute, so on September 23, 2013 we sent a request to ⟨UNIVERSITY⟩'s legal department:

```
Dear Sir/Madam,

My name is Christian Collberg, and I'm a Professor
of Computer Science at the University of Arizona.

Last week I made a  request to Prof. ⟨PROFESSOR⟩ of
the Computer Science Department at ⟨UNIVERSITY⟩.
Please see below. It has been over 3 days, and I have
yet to receive a response. I wonder if you could let me
know what further steps I can take.
```

On September 24, 2013 we received the following response:

```
Dear Dr. Colberg:

The Office of Legal Affairs at ⟨UNIVERSITY⟩ has
received your Open Records Act Request, dated September 16, 2013,
appearing below.  We understand your request to be seeking:

  "ALL SOURCE CODE, NOTES, AND TEST DATA
   RELATED TO THE ⟨SYSTEM-R⟩ SYSTEM"

The requested records have not been located.  We are still in the
process of checking with additional personnel within the ⟨UNIVERSITY⟩.
However, to the extent such records may exist, they will not be
produced pursuant to ⟨OPEN RECORDS STATUTE⟩.  Please let me know if you have any
questions or concerns.  We will let you know if there are any such
records in existence as soon as possible.  Thank you.
```

The provision of the ⟨OPEN RECORDS STATUTE⟩ the preceding email refers to states that:

```
  Any data, records, or information developed, collected, or
  received by or on behalf of faculty, staff, employees, or students
  of an institution of higher education or any public or private
  entity supporting or participating in the activities of an
  institution of higher education in the conduct of, or as a result
  of, study or research on medical, scientific, technical, scholarly,
  or artistic issues, whether sponsored by the institution alone or in
  conjunction with a governmental body or private entity, until such
  information is published, patented, otherwise publicly disseminated,
  or released to an agency whereupon the request must be made to the
  agency. This paragraph shall apply to, but shall not be limited to,
  information provided by participants in research, research notes and
  data, discoveries, research projects, methodologies, protocols, and
  creative works;
```

On September 24, 2013, we replied to the ⟨UNIVERSITY⟩ legal department with the following email:

I am confused by your statement

> "We are still in the process of checking with additional personnel within the ⟨UNIVERSITY⟩. However, to the extent such records may exist, they will not be produced pursuant to ⟨OPEN RECORDS STATUTE⟩."

⟨OPEN RECORDS STATUTE⟩ states that:

> Any data, records, or information developed, collected, or received by or on behalf of faculty, staff, employees, or students of an institution of higher education or any public or private entity supporting or participating in the activities of an institution of higher education in the conduct of, or as a result of, study or research on medical, scientific, technical, scholarly, or artistic issues, whether sponsored by the institution alone or in conjunction with a governmental body or private entity, until such information is published, patented, otherwise publicly disseminated, or released to an agency whereupon the request must be made to the agency. This paragraph shall apply to, but shall not be limited to, information provided by participants in research, research notes and data, discoveries, research projects, methodologies, protocols, and creative works;

Since the information we're requesting has been published the clause "until such information is published, patented, otherwise publicly disseminated" applies, and the information should be released. The clause "or released to an agency whereupon the request must be made to the agency" does not apply since what was released to ⟨MILITARY AGENCY⟩ appears to be the following technical report

> ⟨FINAL REPORT RELEASED TO MILITARY AGENCY⟩

which does not include the source code of the ⟨SYSTEM-R⟩ system, which we are requesting, and that information should be released.

I would appreciate it if you would clear up these issues for me. Also, please let me know what steps you are taking to locate the requested information and when you believe this investigation will have been concluded.

On September 26, 2013, we received this response from ⟨UNIVERSITY⟩'s legal department:

Good afternoon,

We have been unable to locate a confirmed instance of ⟨SYSTEM-R⟩ source code on any ⟨UNIVERSITY⟩ system. The requested records do not exist.

Please let me know if you have any additional questions or concerns.

A.2.4 Formal Request for Email Trail

We next sent a request for the emails between the authors, in order to trace the whereabouts of the source code of ⟨SYSTEM-R⟩:

```
Pursuant to ⟨OPEN RECORDS STATUTE⟩, the Open Records Act,
I request copies of all electronic mail between
   ⟨MAIN STUDENT⟩
   ⟨SECOND STUDENT⟩
   ⟨JUNIOR-PROFESSOR⟩
   ⟨PROFESSOR⟩
regarding the article
    ⟨PAPER⟩
published in
     ⟨CONFERENCE⟩
and the development of the
    ⟨SYSTEM-R⟩
system described therein.

In accordance with ⟨OPEN RECORDS STATUTE⟩ I request that the
electronic databases to which the following email addresses
are attached be searched:
    ⟨UNIVERSITY EMAILA ADDRESSES⟩.
I also request that any private (non-university) email accounts containing
work-related emails be searched.

I am willing to pay applicable fees. These records are sought in
furtherance of scholarly research, and I am employed by an Educational,
Non-commercial Scientific Institution. Therefore, I ask that fees,
other than duplication fees, be waived.

CC:
    ⟨DEPARTMENT CHAIR⟩
```

We received this reply:

Good morning,

The Office of Legal Affairs at ⟨UNIVERSITY⟩ has
received your ⟨OPEN RECORDS STATUTE⟩ request appearing below. We understand
your request to be seeking:

all electronic mail between
⟨MAIN STUDENT⟩
⟨SECOND STUDENT⟩
⟨JUNIOR-PROFESSOR⟩
⟨PROFESSOR⟩
regarding the article
⟨PAPER⟩
published in
⟨CONFERENCE⟩
of the
⟨SYSTEM-R⟩
System [source code] described therein

To the extent that any such records may exist, they will not be produced
pursuant to ⟨OPEN RECORDS STATUTE⟩

⟨PARALEGAL, OFFICE OF LEGAL AFFAIRS⟩

This message is intended to be used exclusively by the addressee(s).
Unauthorized disclosure or use of this information is strictly prohibited.
If you have received this communication in error, please permanently
dispose of the original message and notify me immediately by sending an
email to ⟨EMAIL ADDRESS⟩

Also, please note that most communications to or from ⟨UNIVERSITY⟩
employees are a public record and available to the public and the media
upon request under ⟨STATE⟩'s broad open records law. Therefore, this
e-mail communication and any response may be subject to public disclosure.

We responded:

Thanks for your quick response.

> To the extent that any such records may exist, they will not be produced
> pursuant to ⟨OPEN RECORDS STATUTE⟩.

As you are aware, ⟨OPEN RECORDS STATUTE⟩ state that:

(35) Data, records, or information of a proprietary nature produced or
collected by or for faculty or staff of state institutions of higher
learning, or other governmental agencies, in the conduct of, or as a
result of, study or research on commercial, scientific, technical, or
scholarly issues, whether sponsored by the institution alone or in
conjunction with a governmental body or private concern, where such
data, records, or information has not been publicly released,
published, copyrighted, or patented;

(36) Any data, records, or information developed, collected, or
received by or on behalf of faculty, staff, employees, or students of
an institution of higher education or any public or private entity
supporting or participating in the activities of an institution of
higher education in the conduct of, or as a result of, study or
research on medical, scientific, technical, scholarly, or artistic
issues, whether sponsored by the institution alone or in conjunction
with a governmental body or private entity, until such information is
published, patented, otherwise publicly disseminated, or released to
an agency whereupon the request must be made to the agency. This
paragraph shall apply to, but shall not be limited to, information
provided by participants in research, research notes and data,
discoveries, research projects, methodologies, protocols, and creative
works;

The clause "information has not been publicly released, published,
copyrighted, or patented" is relevant here, since the ⟨SYSTEM-R⟩ system
has been published. Hence, information about this system, such as
research notes, contained, as in this case, in emails, is public
record and subject to disclosure. The clause "released to an agency
whereupon the request must be made to the agency" does *not* apply
since, what was released to the NSF/⟨MILITARY AGENCY⟩ was research reports, and not
source code and research notes.

I must therefore reiterate my request for emails between the authors
related to the ⟨SYSTEM-R⟩ system to be released.

> Also, please note that most communications to or from ⟨UNIVERSITY⟩
> employees are a public record and available to the public and the media
> upon request under ⟨STATE⟩'s broad open records law. Therefore, this
> e-mail communication and any response may be subject to public disclosure.

:)

Best wishes,
Christian Collberg

On October 8, 2013, we received the final communication:

> Dr. Collberg:
>
> The Office of Legal Affairs at ⟨UNIVERSITY⟩ has
> received your renewed/reiterated Open Records Act request below, dated
> October 3, 2013. We understand your request to be seeking emails
> to/from ⟨PROFESSOR⟩, ⟨MAIN STUDENT⟩, ⟨SECOND STUDENT⟩, ⟨JUNIOR-PROFESSOR⟩
> ‘‘related to the ⟨SYSTEM-R⟩ system’’.
>
> To the extent that responsive records may exist, we estimate a total
> cost of $2,263.66* to search for, retrieve, redact and produce such
> records. There will also be a charge of $.10 per copy for any paper
> copies that you would like to receive. Please note that we will not
> produce any records, or portions of records, that are exempted
> pursuant to ⟨OPEN RECORDS STATUTE⟩, including but not limited to ⟨subclause⟩.
>
> If you would like to proceed with your request, please provide your
> check payable to ‘‘⟨UNIVERSITY⟩’’ for $2,263.66.
> Please note that you will be charged the actual cost, regardless of
> whether it is more or less than the estimated amount.
>
> **8 hrs @ $34.04 search/retrieval
> 37 hrs @53.82 review/redact records
>
> ⟨PARALEGAL, OFFICE OF LEGAL AFFAIRS⟩

We did not take up the ⟨UNIVERSITY⟩ on their offer, since we believed the charges requested were not reasonable under the ⟨OPEN RECORDS STATUTE⟩, which states:

> (c)(1) An agency may impose a reasonable charge for the search,
> retrieval, redaction, and production or copying costs for the
> production of records pursuant to this article. An agency shall
> utilize the most economical means reasonably calculated to identify
> and produce responsive, non excluded documents. Where fees for
> certified copies or other copies or records are specifically
> authorized or otherwise prescribed by law, such specific fee shall
> apply when certified copies or other records to which a specific fee
> may apply are sought. In all other instances, the charge for the
> search, retrieval, or redaction of records shall not exceed the
> prorated hourly salary of the lowest paid full-time employee who, in
> the reasonable discretion of the custodian of the records, has the
> necessary skill and training to perform the request; provided,
> however, that no charge shall be made for the first quarter hour.

A.2.5 Request for the NSF grant application

We made a formal request to the NSF for the two applications of the grants that supported the research. In one, the Principal Investigator (⟨PROFESSOR⟩ in the discussion above) writes:

```
We wil also make our data and software available to the research
community when appropriate.
```

Needless to say, we never received any communication from ⟨PROFESSOR⟩, in particular no explanation as to why releasing the code might not have been "appropriate."