An Evaluation of the Ninth SOSP Submissions

or

How (and How Not) to Write a Good Systems Paper

Roy Levin and David D. Redell

Ninth SOSP Program Committee Co-chairmen

Introduction

On March 21, 1983, the program committee for the 9th Symposium on Operating System Principles, having read the eighty-three papers submitted, selected sixteen for presentation at the symposium. This acceptance ratio of about one in five approximates those of past SOSPs, although the number of submissions was somewhat lower than in recent years. Several members of the program committee found it surprisingly easy to separate the good papers from the bad ones; indeed, the ten committee members quickly agreed on the disposition of over 80% of the papers. As the acceptance ratio indicates, most of these were rejections.

After the committee had completed its selection process, several members expressed disappointment in the overall quality of the submissions. Many of the rejected papers exhibited similar weaknesses, weaknesses that the committee felt should have been evident to the authors. In the hope of raising the quality of future SOSP submissions, and systems papers generally, the committee decided to describe the criteria used in evaluating the papers it received. This article combines the criteria used by all of the members of the committee, not just the authors.

To try to avoid sounding preachy or pedagogic, we have cast this presentation in the first and second person and adopted a light, occasionally humorous style. Nevertheless, the intent is serious: to point out the common problems that appear repeatedly in technical papers in a way that will make it easier for future authors to avoid them. As you read this article, then, suppose yourself to be a prospective author for the 10th SOSP or for TOCS. You've done some work you would like to publish, so you sit down to write a paper. What questions should you be asking yourself as you write? These are also the questions that we, the reviewers of your paper, will be asking to determine its suitability for publication.

Classes of Papers

Your paper will probably fall naturally into one of three categories:

- It presents a real system, either by a global survey of an entire system or by a selective examination

Published in the ACM's *Operating Systems Review*, Volume 17, Number 3, July 1983 pages 35 – 40

of specific themes embodied in the system.

- It presents a system that is unimplemented but utilizes ideas or techniques that you feel the technical community should know.
- It addresses a topic in the theoretical areas, for example, performance modelling or security verification.

Obviously, a single set of evaluation criteria cannot be applied uniformly across these categories; nevertheless, many criteria apply equally well to all three. As we describe each one below, we will try to emphasize the classes of papers to which it applies. Often it will be evident from context.

Criteria for Evaluation of Submissions

Original Ideas

Are the ideas in the paper new? There is no point in submitting a paper to a conference or journal concerned with original work unless the paper contains at least one new idea.

How do you know? You must be familiar with the state of the art and current research in the area covered by your paper in order to know that your work is original. Perhaps the most common failing among the submissions in the first category (real systems) was an absence of new ideas; the systems described were frequently isomorphic to one of a small number of pioneering systems well-documented in the literature.

Can you state the new idea concisely? If your paper is to advance the state of knowledge, your reader must be able to find the new ideas and understand them. Try writing each idea down in a paragraph that someone generally versed in the relevant area can understand. If you can't, consider the possibility that you don't really understand the idea yourself. When you have the paragraphs, use them in the abstract for the paper.

What exactly is the problem being solved? Your reader cannot be expected to guess the problem you faced given only a description of the solution. Be specific. Be sure to explain why your problem couldn't be solved just as well by previously published techniques.

Are the ideas significant enough to justify a paper? Frequently, papers describing real systems contain one or two small enhancements of established techniques. The new idea(s) can be described in a few paragraphs; a twenty-page paper is unnecessary and often obscures the actual innovation. Since construction of a real system is a lot of work, the author of the paper sometimes unconsciously confuses the total effort with the work that is actually new. ("My team worked on this system for two years and we're finally done. Let's tell the world how wonderful it is.") If the innovation is small, a small paper or technical note in a suitable journal is more appropriate than an SOSP submission.

Is the work described significantly different from existing related work? An obvious extension to a previously published algorithm, technique, or system, does not generally warrant publication. Of course, the label "obvious" must be applied carefully. (Remember the story of Columbus demonstrating how to make an egg stand on end (by gently crushing it): "it's obvious once I've shown you how".) You must show that your work represents a significant departure from the state of the art. If you can't, you should ask yourself why you are writing the paper and why anyone except your mother should want to read it.

Is all related work referenced, and have you actually read the cited material? You will have difficulty convincing the skeptical reader of the originality of your efforts unless you specifically distinguish it from previously published work. This requires citation. Furthermore, you will find it harder to convince your reader of the superiority of your approach if he has read the cited works and you haven't.

Published in the ACM's *Operating Systems Review*, Volume 17, Number 3, July 1983 pages 35 – 40

Are comparisons with previous work clear and explicit? You cannot simply say: "Our approach differs somewhat from that adopted in the BagOfBits system [3]." Be specific: "Our virtual memory management approach uses magnetic media rather than punched paper tape as in the BagOfBits system [3], with the expected improvements in transfer rate and janitorial costs."

Does the work comprise a significant extension, validation, or repudiation of earlier but unproven ideas? Implementation experiences supporting or contradicting a previously published paper design are extremely valuable and worthy candidates for publication. Designs are cheap, but implementations (particularly those based on unsound designs) are expensive.

What is the oldest paper you referenced? The newest? Have you referenced similar work at another institution? Have you referenced technical reports, unpublished memoranda, personal communications? The answers to these questions help alert you to blind spots in your knowledge or understanding. Frequently, papers with only venerable references repeat recently published work of which the author is unaware. Papers with only recent references often "rediscover" (through ignorance) old ideas. Papers that cite only unpublished or unrefereed material tend to suffer from narrowness and parochialism. Remember that citations not only acknowledge a debt to others, but also serve as an abbreviation mechanism to spare your reader a complete development from first principles. If the reader needs to acquire some of that development, however, he must be able to convert your citations into source material he can read. Personal communications and internal memoranda fail this test. Technical reports are frequently published in limited quantities, out-of-print, and difficult to obtain. Consequently, such citations as source material should be avoided wherever possible.

Reality

Does the paper describe something that has actually been implemented? Quite a few of the SOSP submissions proceeded for fifteen pages in the present tense before revealing, in a concluding section (if at all), that the foregoing description was of a hypothetical system for which implementation was just beginning or being contemplated. This is unacceptable. Your reader has a right to know at the outset whether the system under discussion is real or not.

If the system has been implemented, how has it been used, and what has this usage shown about the practical importance of the ideas? Once again, a multiple man-year implementation effort does not of itself justify publication of a paper. If the implemented system contains new ideas, it is important to explain how they worked out in practice. A seemingly good idea that didn't pan out is at least as interesting as one that did. It is important to be specific and precise. "Our weather prediction system is up and running and no one has complained about its occasional inaccurate forecasts" is much less convincing than "everytime we fail to forecast rain, the users hang their wet shirts over the tape drives to dry". In the latter case, at least we know that people are using and depending on the system.

If the system hasn't been implemented, do the ideas justify publication now? This can be a difficult question for an author to answer dispassionately, yet any reviewer of the paper will make this judgment. It is always tempting to write a design paper describing a new system, then follow it up in a year or two with an "experience" paper. The successful papers of this genre nearly always include initial experience in the closing sections of the design paper. The subsequent experience paper then deals with the lessons learned from longer-term use of the system, frequently in unanticipated ways. Reviewers are very skeptical of design-only papers unless there are new ideas of obviously high quality.

Lessons

What have you learned from the work? If you didn't learn anything, it is a reasonable bet that your readers won't either, and you've simply wasted their time and a few trees by publishing your paper.

What should the reader learn from the paper? Spell out the lessons clearly. Many people repeat the

Published in the ACM's *Operating Systems Review*, Volume 17, Number 3, July 1983 pages 35 – 40

mistakes of history because they didn't understand the history book.

How generally applicable are these lessons? Be sure to state clearly the assumptions on which your conclusions rest. Be careful of generalizations based on lack of knowledge or experience. A particularly common problem in "real system" papers is generalization from a single example, e.g., assuming that all file system directories are implemented by storing the directory in a single file and searching it linearly. When stating your conclusions, it helps to state the assumptions again. The reader may not have seen them for fifteen pages and may have forgotten them. You may have also.

Choices

What were the alternatives considered at various points, and why were the choices made the way they were? A good paper doesn't just describe, it explains. Telling your readers what you did doesn't give them any idea how carefully considered your choices were. You want to save future researchers from following the same blind alleys. You also want to record potentially interesting side-streets you didn't happen to explore. Make sure to state clearly which is which.

Did the choices turn out to be right, and, if so, was it for the reasons that motivated them in the first place? If not, what lessons have you learned from the experience? How often have you found yourself saying "this works, but for the wrong reason"? Such a pronouncement represents wisdom (at least a small amount) that may benefit your reader. Many papers present a rational argument from initial assumptions all the way to the finished result when, in fact, the result was obtained by an entirely different path and the deductive argument fashioned later. This kind of "revisionist history" borders on dishonesty and prevents your readers from understanding how research really works.

Context

What are the assumptions on which the work is based? The skeptical reader is unlikely to accept your arguments unless their premises are stated. Make sure you get them all; it's easy to overlook implicit assumptions.

Are they realistic? For "unimplemented systems" papers, this amounts to asking whether the assumptions of the design can hope to support a successful implementation. Many paper designs are naive about the real characteristics of components they treat abstractly, e.g., communication networks or humans typing on terminals. For theoretical studies, it must be clear how the assumptions reflect reality, e.g., failure modes in reliability modelling, classes of security threats in security verification, arrival distributions in queuing systems.

How sensitive is the work to perturbations of these assumptions? If your result is delicately poised on a tall tower of fragile assumptions, it will be less useful to a reader than one that rests on a broader and firmer foundation.

If a formal model is presented, does it give new information and insights? Simply defining a model for its own sake is not very useful. One deep theorem is worth a thousand definitions.

Focus

Does the introductory material contain excess baggage not needed for your main development? "Real system" papers are particularly guilty of irrelevant description. If your subject is distributed file systems, the physical characteristics of the connection between computer and communication network are probably not germane. Avoid the temptation to describe all major characteristics of your system at the same level of depth. Concentrate instead on the novel or unusual ones that (presumably) will be the focus of the original technical content of the paper.

Do you include just enough material from previously published works to enable your reader to follow

Published in the ACM's *Operating Systems Review*, Volume 17, Number 3, July 1983 pages 35 – 40

your thread of argument? Do not assume that the reader has read every referenced paper within the last week and has them at his fingertips for instant reference. If you want your reader to get past page three, avoid introductory sentences of the form "We adopt the definition of transactions from Brown [4], layering it onto files as described by Green [7, 18], with the notions of record and database introduced by Black [10] and White [12] and later modified by Gray [6]". On the other hand, don't burden your reader unnecessarily with lengthy extracts or paraphrases from cited works.

Presentation

Are the ideas organized and presented in a clear and logical way?

Are terms defined before they are used?

Are forward references kept to a minimum? Readers get annoyed when they repeatedly encounter statements like "Each file consists of a sequence of items, which will be described in detail in a later section". The reader has to remember the technical term "item", but the term has no semantics yet. It's all right to ask him to do this once or twice, but only when absolutely necessary. Even if you can't afford the digression to explain "item" at this point, give the reader enough information to attach some meaning to the term: "Each file consists of a sequence of items, variable-sized, self-identifying bit sequences whose detailed interpretation will be discussed below under 'Multi-media Files'." Your reader may not yet understand your concept of files completely, but at least he has some glimpse of the direction in which you are leading him.

Have alternate organizations been considered? Theoretical papers, particularly of a mathematical character, are generally easier to organize than papers describing systems. The expected sequence of definition, lemma, theorem, example, corollary works well for deductive argument, but poorly for description. In "real system" papers, much depends on the intent: global survey or selective treatment. Frequently, difficulties in organization result from the author's unwillingness to commit to either approach. Decide whether you are surveying your system or focusing on a specific aspect and structure the paper accordingly.

Was an abstract written first? Does it communicate the important ideas of the paper? Abstracts in papers describing systems are sorely abused. The abstract is more often a prose table of contents than a precis of the technical content of the paper. It tends to come out something like this: "A system based on Keysworth's conceptualization of user interaction [4] has been designed and implemented. Some preliminary results are presented and directions for future work considered." No reader skimming a journal is likely to keep reading after that. Avoid the passive voice (despite tradition) and include a simple statement of assumptions and results. "We designed and implemented a user interface following the ideas of Keysworth and discovered that converting the space bar to a toe pedal increases typing speed by 15%. However, accuracy decreased dramatically when we piped rock music instead of Muzak^{TM} into the office." Leave discussion and argument for the paper. It helps to write the abstract before the paper (despite tradition) and even the outline, since it focusses your attention on the main ideas you wants to convey.

Is the paper finished? Reviewers can often help you to improve your paper, but they can't write it for you. Moreover, they can't be expected to interpolate in sections marked "to be included in the final draft". In a mathematical paper, a reviewer regards the statement of a theorem without proof with suspicion, and, if the theorem is intended to culminate prior development, with intolerance. Similarly, in a paper describing a system, a reviewer cannot tolerate the omission of important explanation or justification. Omitting sections with a promise to fill them in later is generally unacceptable.

Writing Style

Is the writing clear and concise?

Published in the ACM's *Operating Systems Review*, Volume 17, Number 3, July 1983 pages 35 – 40

Are words spelled and used correctly?

Are the sentences complete and grammatically correct?

Are ambiguity, slang, and cuteness avoided?

If you don't have sufficient concern for your material to correct errors in grammar, spelling, and usage before submitting it for publication, why should you expect a reviewer to read the paper carefully? Some reviewers feel that this kind of carelessness is unlikely to be confined to the presentation, and will reject the paper at the first inkling of technical incoherence. Remember that you are asking a favor of your reviewers: "Please let me convince you that I have done interesting, publishable work." A reviewer is more favorably disposed toward you if he receives a clean, clear, carefully corrected manuscript than if it arrives on odd-sized paper after ten trips through a photocopier and looking like it was composed by a grade-school dropout. Even if you aren't particularly concerned with precise exposition, there is certain to be someone in your organization who is. Give your manuscript to this conscientious soul and heed the resulting suggestions.

Summary

These thirty-odd questions can help you write a better technical paper. Consult them often as you organize your presentation, write your first draft, and refine your manuscript into its final form. Some of these questions address specific problems in "systems" papers; others apply to technical papers in general. Writing a good paper is hard work, but you will be rewarded by a broader distribution and greater understanding of your ideas within the community of journal and proceedings readers.

This article first appeared in ACM SIGOPS Operating Systems Review, Vol. 17, No. 3 (July, 1983), pages 35-40. Every effort has been made to keep the text and formatting in this document identical to that of the original article. The text in this file was scanned using OCR technology and has been carefully proofread, but some scanning errors may remain. This document is being made available with the permission of Roy Levin, one of the authors.

Shawn Ostermann
Mon Oct 2, 1995