The Many Faces of Systems Research – And How to Evaluate Them

Aaron B. Brown, *IBM Research*
Anupam Chanda, *Rice University*
Rik Farrow, *Consultant*
Alexandra Fedorova, *Harvard University*
Petros Maniatis, *Intel Research*
Michael L. Scott, *University of Rochester*

Abstract

Improper evaluation of systems papers may result in the loss or delay in publication of possibly important research. This paper posits that systems research papers may be evaluated in one or more of the three dimensions of science, engineering and art. Examples of these dimensions are provided, and methods for evaluating papers based on these dimensions are suggested. In the dimension of science, papers can be judged by how well they actually follow the scientific method, and by the inclusion of proofs or statistical measures of the significance of results. In engineering, the applicability and utility of the research in solving real world problems is the main metric. Finally, we argue that art be considered as a paper category evaluated based on elegance, simplicity, and beauty.

1 Introduction

Evaluating systems research[1] is hard. Systems research is multifaceted; it often involves proving scientific hypotheses as well as designing and implementing real systems. As such, it goes beyond traditional science, spreading into the realm of engineering and perhaps even art, as system designers strive for elegance in their systems' blueprints. In this paper, we argue that evaluation criteria for systems research should match the dimension — engineering, science, art — in which particular work extends.

Every systems project maps to different points on each of these dimensions. A study evaluating performance of a new system could be regarded as "engineering" research. However, the ultimate goal of a performance study is typically to prove or disprove a hypothesis, to find the "truth," and this manifests the scientific dimension of the study. Likewise, research that might at first be classified as "art" can sometimes contribute to "science" or "engineering" as well, particularly if it introduces a new perspective that simplifies scientific understanding or improves ease of use. Failure to recognize the multidimensionality of systems research leads to subjective evaluation and misuse of evaluation criteria—for example engineering metrics are used to evaluate an "artistic" or a "scientific" research result.

In the rest of the paper we describe in more detail these research categories, partly prescriptively by specifying the qualities that exemplars of each dimension exhibit, and by example by pointing out specific instances of each (Section 2). In Section 3, we expand on the evaluation criteria that appear appropriate for each dimension, drawing from the non-CS disciplines after which each dimension is modelled. In Section 4 we propose an action plan for the improvement of systems research evaluation based on these three dimensions. Finally, we present related work in Section 5 and then conclude.

2 Dimensions of Systems Research

In this section, we identify the driving forces within the evaluation dimensions of science, engineering, and art. We also describe instances whose principal components exemplify individual or combined dimensions.

2.1 Science

Ironically, not many obvious opportunities exist readily in Computer Science to conduct Science. Computer artifacts are human-made; they are not natural parts of the physical world and, as such, have few laws to be discovered that computer engineers did not themselves instill into the field (although this perspective is slowly changing as systems grow so complex that they begin to exhibit emergent behavior). And yet, computer science deals with computers, which typically run with electrons carrying information bits, which in turn are governed by the laws of physics as much as any other aspect of the physical world. In the traditional terminology of databases, computer science works on a view of the ground truths of the universe, as manipulated by com-

puter architecture, by software, and by the applications that business, "mainstream" science, and entertainment have demanded. In this manner, science in Computer Science deals with studying the manifestation of the laws of the universe in the artifacts of the field. Consequently, "scientific" systems research strives to discover truth, by forming hypotheses and then proving or disproving those hypotheses mathematically or via experimentation, as well as identifying the effects and limitations of computer artifacts on the physical world.

A typical scientific example from the systems literature proves the impossibility of distributed consensus in asynchronous systems under faults [10]. Given a model of asynchronous communication, the authors show that consensus cannot be achieved when even one participant fails. The truth of the result, under the starting conditions of the analysis, is indisputable and relies on mathematical logic. Engineering, however (see below), can make use of this truth as necessary; for instance, an actual protocol or application that ensures it does not fall under the model of the impossibility result can have hopes of achieving distributed consensus.

In some cases, stepping back and looking at the behavior of large populations of human-made artifacts in the proper scope can yield to scientific study of a manner similar to that employed in physical sciences. Of particular importance in the networking community has been the identification of self-similarity in network traffic [15]. This work establishes that network packet traffic has self-similar characteristics, by performing a thorough statistical analysis of a very large set of Ethernet packet traces. Before this work was published, "truth" was that network traffic could be modeled as a Poisson process, deeply affecting every analysis of networks. For example, in the previous model, traffic aggregation was thought to help "smooth" bursts in traffic, which conflicted with practical observation; this work identified the reasons for this conflict.

Finally, a third example demonstrates truth at the boundaries of engineering, by identifying structure within artifacts imposed on the physical world. Work on the duality of operating systems structures [14] proposes a fundamental set of principles mapping between two competing system engineering disciplines, and demonstrates a fundamental "truth": no matter how you slice them, message-oriented systems and procedure-oriented systems are equivalent and can only be differentiated in terms of engineering convenience, not by their inherent strengths or weaknesses compared to each other. Instead of discovering a truth that was out there, this work looks at the effects of design choices to the physical world and identifies structure within them. In this case, the discovered structure is a duality.

2.2 Engineering

Whereas science seeks to uncover truth, regardless of where that truth resides or how it can be brought to bear on practice, engineering starts with the inevitability of practical relevance and goes backward to the principles that make practical utility achievable. In the example of distributed consensus above, science demanded that a universally true statement be made. However, where does this statement leave the systems researcher who still needs to deal with fault-tolerant consensus in practical systems? The answer lies in the engineering pursuit of achieving a solution that works for a particular problem and might not, necessarily, generalize; it lies as well in the analysis of such heuristics to understand when they are "good enough" for practical use. For example, Castro and Liskov [8] employ a type of weak synchrony to exit the asynchronous regime of [10] and, thereby, achieve fault-tolerant, distributed consensus for replicated services. A practical engineering decision was made (imposing some restrictions on how communication is conducted) to enable a solution for a real problem.

Unlike this example, in many cases, engineering systems research presents no new truths; it deals with solving a particular problem by synthesizing truths and solutions previously proposed. When the problem is shared by a large population, the utility of such a solution can outstrip significantly the utility of a new truth, at least in the short term. The Google File System, for instance [11], is a layer that supports the Internet searches of millions of users every day. Its authors admit that its design is not intended to be general or even applicable to any other storage problem. Yet, the broad relevance of the target application makes this engineering effort worthwhile and significant.

2.3 Art

In systems research, art has been a controversial evaluation dimension. Its manifestations as elegance, or, when stretched to its subjective extreme, beauty, can make complex ideas more palatable or more comprehensible. Elegance, as economy of expression in system abstractions, interfaces, and languages may help to sell the argument behind a complex idea [20], or bring order in an area where chaos reigned before. It it also a key contributor to the user experience for computer systems, affecting long-term ownership and maintenance costs: systems with elegance in their underlying designs are often easier to use and manage as well as being less prone to human error. On the other hand, elegance is sometimes parsimonious, economic but hard to comprehend [5]. Finally, often artful system design is its own goal.

The classic THE multiprogramming system [9] can be considered an example of elegant, simple, harmonious system design. Layering is pushed to the extreme, pro-

viding for a clean separation of concerns and a design that promotes composable verification of individual system components. Though an early version of a complex software system, this work exemplified the beauty and elegance of clean interfaces to enhance system understandability. In a more general way, BAN logic [7] introduced a simple, contained structure to reasoning about authentication protocols, shaping an entire field for years to come. On the side of engineering elegance, Tangler [19] is a collaborative storage system that balances function with participation incentives. To ensure his documents are protected against censorship a user must employ and remember a source of randomness found in another user's documents. Incentives for storing foreign content are balanced by a user's need to retrieve his own content.

As in all art, the beauty in elegant system research often lies in the (subjective) eye of the beholder. In the examples above, the THE system sacrificed efficiency for strict layering, BAN logic lacked an elegant path for adoption by error-prone humans, and Tangler was a niche application.

2.4 Discussion

One could argue that influential systems research scores high on multiple dimensions. An elegant and scientifically sound study is strictly better (more understandable, more extensible, etc.) than a sterile, unintuitive yet correct study. For example, Gummadi et al. [12] present a great example of engineering elegance, by abstracting away implementation-specific details of different distributed hash table algorithms and distilling a simple engineering rule for the selection of the geometry of overlays. On the other hand, elegance that is patently false is often the weapon of a demagogue. As we illustrate in the following section, when evaluating systems research, elegance cannot replace correctness.

3 Evaluation Criteria

Each of the dimensions introduced in the previous section requires its own set of evaluation criteria. In this section we develop these criteria, treating the dimensions as independent; work that spans multiple dimensions should be assessed on the aggregate criteria of all dimensions touched.

3.1 Science

The value of systems work that falls into the science category lies in its ability to expose new truths about the constructs, abstractions, architectures, algorithms, and implementation techniques that make up the core of systems research. On one hand, evaluation criteria must fathom the depth of the truths revealed, based on novelty, their ability to categorize and explain existing constructs or behaviors, and their generality and applicability to multiple platforms and environments.

Papers that build a comprehensive design space around a set of ad-hoc techniques (like peer-to-peer routing protocols), or those that provide a foundation for understanding the tradeoffs in the use of different constructs (like threads vs. events, VMMs vs. microkernels), are stronger science than those that simply provide engineering insight into the behavior of a particular system implementation.

On the other hand, it is important to evaluate the methodological rigor of systems science. The essence of science — the scientific method — involves the careful testing or mathematical proof of an explicitly-formed hypothesis. Evaluators of science should look for work that forms a clear hypothesis, that constructs reproducible experiments to shed light on that hypothesis while controlling other variables, and that includes the analysis needed to prove or disprove the hypothesis. In particular, careful measurements are essential to strong science work.

3.2 Engineering

The value of systems work that falls into the engineering category lies in its utility: the breadth of applicability of the engineering technique to important real-world contexts, and the power of the technique to solve important problems in those contexts. Engineering work that succeeds on the first criterion will define techniques that open up a broad space of new applications—such as the introduction of BPF [17], which enabled a large body of work on such varied topics as intrusion detection, worm filtering, and tunneling—or that addresses a persistent problem that appears in many contexts, such as caching. Work that succeeds on the second criterion will typically introduce a method for solving a problem that is more effective than any existing known solution—a "best of breed" technique. The best engineering work succeeds on both criteria, introducing powerful solutions to broadly-applicable problems. Work that only addresses one criterion must be examined carefully relative to its practical utility; for example, work that provides a powerful solution to a non-problem ("engineering for its own sake") does not represent high-value engineering research, although it might fall into the category of art.

Another key criterion for evaluating engineering work, especially in the form of a research paper, is the strength of its evaluation. Good engineering work includes detailed measurements that demonstrate the value of the work along both the power and applicability axes. The latter is key—a paper that claims broad applicability but only measures its technique on a series of microbenchmarks does not demonstrate its value as well as one that analyzes the technique's power across a variety of realistic environments.

3.3 Art

Evaluating systems research that falls into the art category is inherently a subjective business. The typical evaluation criteria for art include elegance, beauty, simplicity, and its ability to introduce new perspectives on existing, well-trodden areas. None of these are easily quantified except perhaps the last, and even that is subject to interpretation. Human factors studies (such as those commonly found in HCI research) are perhaps a first step toward collecting and correlating evaluations of aspects of art, such as usability or elegance, and perhaps should be seen more frequently in systems research work; however, they remain based on subjective assessments. Thus, since there will likely always be some disagreement about artistic value, "artful" systems research is best left to be evaluated by its consumers and the impartial view of history. A more practical approach, perhaps, is to use a panel of expert judges as is done in other artistic fields; the existing construct of a program committee fits well into this paradigm although it can suffer the same capriciousness that plagues judging in the arts.

4 A Prescription for More Rigorous Evaluation

While the systems research community has an excellent track record of producing high-quality, high-impact research in all three dimensions of science, engineering, and art, it has historically fallen short in evaluating that work with the rigor and discipline of other scientific and engineering communities. This weakness can be attributed to many factors, including the field's relative youth and its tight association with the fast-moving marketplace, but two stand out in particular: (1) a lack of solid methodology for scientific and engineering evaluation, and (2) a lack of recognition that some systems work is art and must be evaluated as such. As an impetus to remedy this situation, the following sections propose guidelines and research directions to help steer the community toward more rigorous and effective evaluation.

4.1 Science: Revive the Scientific Method

Science is defined by the scientific method, namely the identification of a hypothesis, reproducible collection of experimental data related to that hypothesis, and analysis of the data to evaluate the validity of the hypothesis. Systems research that falls along the science dimension must be evaluated with respect to how well it implements the scientific method.

For the researcher, that means several things. Most important is establishing a well-defined hypothesis. At one extreme, this could be a theorem to be proven; it could also be a claim about system behavior, for example that the same scalability is achievable with threaded architectures as with event-driven architectures. The hypothesis must then be followed up with a set of well-designed, reproducible experiments that illuminate the hypothesis and control for unrelated variables. When control is not possible, enough data must be collected to allow a statistical analysis of the effect of the uncontrolled variables. For example, in the threads-vs.-events hypothesis above, a good set of experiments will control the application, platform, workload, and quality of the implementations being evaluated; if the implementation quality could not be controlled, the experiments should collect data on multiple implementations.

Finally, good science-style systems research must include a sound analysis of the experimental data relative to the hypothesis. A key aspect missing from much systems research is the use of statistics and statistical tests to analyze experimental data—just compare the typical paper in the biological or physical sciences to the typical systems research paper to see the difference! Systems researchers should learn and use the toolbox of statistical tests available to them; systems papers should start reporting p-values to support claims that experimental data proves a hypothesis.

And those evaluating completed systems work—like program committees—should look for and insist on rigorous application of the scientific method, including well-defined hypotheses, reproducible experiments, and the kind of rigorous statistical analysis that we advocate. They should look for experiments and data that directly assess the hypothesis, not just that provide numbers—there are many hypotheses in systems research, particularly in the new focus areas of dependability and reliability, that are not proven by lists of performance figures. Since reproducibility is a key aspect of the scientific method, the community should also provide forums for publishing reproductions of key system results—perhaps as part of graduate student training or in special sessions at key conferences and workshops.

4.2 Engineering: Focus on Real-World Utility

As described in Section 3, the key criterion for evaluating engineering work is applicability. For the researcher, this means that good engineering systems work (and the papers that describe it) will include evaluations illustrating the work's utility in real-world situations. This is a challenge for much modern systems research, since our evaluation metrics and methodologies are primarily built around performance assessment, and the utility challenges faced in many real environments center on other aspects like dependability, maintainability, usability, predictability, and cost. Another challenge is that it is often impractical to evaluate research work directly in the context of real-world deployments or laboratory mock-ups of such systems, so surrogate environments,

such as those defined by application benchmarks, must be used instead.

Thus there are two critical research challenges that must be addressed before we can easily evaluate systems research as to real-world utility. The first involves creating the surrogate environments that researchers, particularly academic researchers, can use to recreate real-world problems and demonstrate the applicability of new engineered technology. Accomplishing this goal will require increased cooperation between industry and academia, and in particular finding ways to transfer the applications and technology behind real-world systems to the academic community. We believe this to be a priority for the continued success and relevance of the systems research community, and call on industry leaders to find solutions to the current stumbling blocks of intellectual property restrictions and licensing costs.

Another promising possibility for creating surrogate environments is to explore ways to do for core systems research what PlanetLab did for distributed and networked computing research: create a shared, commonly-available, realistic mock-up of the complex deployments found in real production environments. For example, a consortium of researchers (academic and industrial) could be assembled to build and operate a production-grade enterprise service (such as a supply-chain operation or an online multiplayer game), providing a test bed for evaluating systems research technologies in the resiliency, security, and maintainability spaces, while sharing the burden of constructing and operating the environment.

The second challenge is to significantly advance our ability to evaluate aspects of utility other than performance. This implies research into metrics, reproducible methodologies, and realistic workloads—benchmarks—for non-performance aspects of systems engineering. Initial work has begun on benchmarks for dependability and usability [6, 13], but much additional research is needed. This effort will require research along all three dimensions of systems research—art to figure out how to approach the problem, science to develop and test methodologies and metrics, and engineering to implement them as benchmarks—and its success will be evaluated by the improvements in rigor and applicability we can achieve in assessing the real-world utility of engineering-style systems research.

4.3 Art: Legitimize Artistic Research

Despite the inherent subjectivity in its evaluation, "artistic" systems research can have tremendous value, particularly in spurring the development of new subfields and in laying the foundation for future advances in science and engineering. But because this value is judged subjectively, artistic research cannot (and should not) be evaluated on the same scale as scientific- or engineering-style research—i.e., one should not demand quantitative results in a paper whose primary contribution is artistic. Today, most forums for publication and discussion receive a mix of artistic, scientific, and engineering submissions, biasing the selection away from the necessarily less-quantified artistic submissions. Instead, the systems community needs to create spaces where artistic research can be presented, examined, and subjectively judged against other artistic research. How this is best accomplished is an open question, since there is a definite risk of marginalizing art papers if handled incorrectly, but one possibility is to dedicate a certain fraction of the paper slots or sessions at the major conferences (or issues of the major journals) to artistic research, perhaps even with a separate reviewing process than the remainder of the conference or journal.

5 Related Work

Related work in this area is available mostly in the form of referee guidelines published by conference program committees and paper evaluation criteria presented in calls for papers. Guidelines for conference referees usually ask committee members to evaluate the degree of technical contribution, novelty, originality and importance to the community [1, 2]. A typical call for papers suggests that a good systems paper would have attacked a significant problem, demonstrated advancement beyond previous work, devised a clever solution and argued its practicality, and drawn appropriate conclusions [3, 4]. Their proposed criteria are overly general and may not fit all types of systems project equally well.

Patterson suggested that the principal criterion for evaluating research is its long-term impact on the technology [18]. While this is a reasonable criterion for a long-running project, it cannot be easily applied to new research, because it is difficult to envision the long-term impact that this research will produce.

Work by Levin and Redell, Ninth SOSP Committee co-chairmen, is perhaps the closest to our work, and is one of the first publications describing a systematic process of evaluating systems research [16]. Like us, they state that there exist different classes of research and that different criteria should be applied for different classes. They propose the following evaluation criteria: originality of ideas, availability of real implementations, importance of lessons learned, extent to which alternative design choices were explored and soundness of assumptions. They describe how to apply these criteria and emphasize which criteria are more appropriate for a particular type of research. In contrast, the contribution of our work is categorization of criteria along the dimensions of science, engineering, and art, as well as the description

of criteria for each dimension and suggestions on how to incorporate those in the evaluation of systems research.

6 Conclusions

Systems research is difficult to evaluate because of its multidimensional nature. In this paper we have identified three dimensions of systems research: science, engineering, and art. We mention several research papers in each of these three domains. For each of these domains we have outlined desirable characteristics to conduct and present research work. Because of the multidimensional nature of systems research, we argue for dimension-specific evaluation criteria. In this regard, we suggest a set of evaluation guidelines for the above mentioned three dimensions. We propose that scientific research works be evaluated by how strictly they adhere to the rigors of scientific methodology; that utility and applicability be the yardstick for engineering research works; and that, in the category of art, research works be judged by their elegance and simplicity. By guiding researchers to better conduct and present their work, and reviewers to evaluate publications with applicable criteria, we believe that this discussion may prove beneficial in improving the systems research landscape.

7 Acknowledgements

The material in this paper was initially developed during a breakout session at the Tenth Workshop on Hot Topics in Operating Systems (HotOS-X). We wish to recognize and thank the following participants as key contributors to the ideas and content of this paper: Pei Cao, Ira Cohen, Robert Grimm, Gernot Heiser, Sharon Perl, Ion Stoica, and Xiaoyun Zhu.

References

[1] 1999 PPoPP electronic referee's report form. `http://www.cs.utah.edu/~wilson/compilers/review-form1.txt`, 1999.

[2] PLDI 2001 referee's report form. `http://www.cs.utah.edu/~wilson/compilers/review-form2.txt`, 2001.

[3] USENIX 2002 call for papers. `http://www.usenix.org/events/usenix02/cfp/usenix02cfp.pdf`, 2002.

[4] OSDI 2004 call for papers. `http://www.usenix.org/events/osdi04/cfp/`, 2004.

[5] BERKHEIMER, G. D., ANDERSON, C. W., AND SPEES, S. T. Using conceptual change research to reason about curriculum. *Research Series No. 195, Michigan State University, Institute for Research on Teaching* (1989).

[6] BROWN, A. B., AND PATTERSON, D. A. Towards availability benchmarks: A case study of software RAID systems. In *USENIX '00: Proceedings of the USENIX Annual Technical Conference* (San Diego, California, USA, June 2000), USENIX Association.

[7] BURROWS, M., ABADI, M., AND NEEDHAM, R. A logic of authentication. *ACM Trans. Comput. Syst. 8*, 1 (1990), 18–36.

[8] CASTRO, M., AND LISKOV, B. Practical byzantine fault tolerance. In *OSDI '99: Proceedings of the third symposium on Operating systems design and implementation* (New Orleans, Louisiana, United States, 1999), USENIX Association, pp. 173–186.

[9] DIJKSTRA, E. W. The structure of the "THE"-multiprogramming system. *Commun. ACM 11*, 5 (1968), 341–346.

[10] FISCHER, M. J., LYNCH, N. A., AND PATERSON, M. S. Impossibility of distributed consensus with one faulty process. *J. ACM 32*, 2 (1985), 374–382.

[11] GHEMAWAT, S., GOBIOFF, H., AND LEUNG, S.-T. The Google file system. In *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles* (Bolton Landing, NY, USA, 2003), ACM Press, pp. 29–43.

[12] GUMMADI, K., GUMMADI, R., GRIBBLE, S., RATNASAMY, S., SHENKER, S., AND STOICA, I. The impact of DHT routing geometry on resilience and proximity. In *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications* (New York, NY, USA, 2003), ACM Press, pp. 381–394.

[13] KANOUN, K., MADIERA, H., AND ARLAT, J. A framework for dependability benchmarking. In *DSN '02: Proceedings of the Workshop on Dependability Benchmarking* (Washington, DC, United States, June 2002).

[14] LAUER, H. C., AND NEEDHAM, R. M. On the duality of operating system structures. *SIGOPS Oper. Syst. Rev. 13*, 2 (1979), 3–19.

[15] LELAND, W. E., TAQQU, M. S., WILLINGER, W., AND WILSON, D. V. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Trans. Netw. 2*, 1 (1994), 1–15.

[16] LEVIN, R., AND REDELL, D. D. An Evaluation of the Ninth SOSP Submissions or How (and How Not) to Write a Good Systems Paper. `http://www.usenix.org/events/samples/submit/advice.html`, 1983.

[17] MCCANNE, S., AND JACOBSON, V. The BSD packet filter: A new architecture for user-level packet filtering. In *USENIX '93: Proceedings of the USENIX Annual Technical Conference* (San Diego, California, USA, Jan. 1993), USENIX Association.

[18] PATTERSON, D. How to have a bad career in research/academia. `http://www.cs.berkeley.edu/~pattrsn/talks/nontech.html`.

[19] WALDMAN, M., AND MAZIERES, D. Tangler: a censorship-resistant publishing system based on document entanglements. In *CCS '01: Proceedings of the 8th ACM conference on Computer and Communications Security* (New York, NY, USA, 2001), ACM Press, pp. 126–135.

[20] WILLIAMS, R. Context, Content and Commodities: e-Learning Objects. In *Proceedings of the 2003 European Conference on eLearning* (Glasgow, UK, Nov. 2003), pp. 485–494.

Notes

[1] Under the term "systems research" we bundle any work that would come out of a "systems group" at a research university, including not only Operating Systems, but networking, distributed systems, theory about systems, etc. In short, we consider work that would conceivably appear in the proceedings of HotOS, OSDI, NSDI, SOSP, etc.