Publication Culture in Computer Science and Software Engineering

Richard P. Gabriel
IBM Research
rpg@dreamsongs.com / rpg@us.ibm.com

I have been involved in the publication of computer science and software engineering papers since the early 1980s. I care very deeply about the quality of writing in our publications, and this has drawn me to work in, work at, and examine the problem of publication from many viewpoints and workpoints. Moreover, I am deeply worried about the question of how ideas can be planted and nurtured in the face of more and more difficult and in a sense more restrictive publication venues—this despite the fact that the past decade has seen an explosion of CS&SE conferences.

Richard P. Gabriel's Relevant Experience

1986, Lisp & Functional Programming: I was general chair for this conference. At that time the conference accepted practical and engineering papers—experience reports, implementation papers, language design papers—alongside theoretical papers. In 1984 I had three papers in L&FP, and in 1986, 1988, 1990, and 1992 I was in the inner circle of people influencing this conference. In the mid-1990s the conference merged with Functional Programming and Computer Architecture (FPCA) to become the International Conference on Functional Programming, and soon it became a theory conference with few or no engineering papers accepted.

1988, Lisp and Symbolic Computation: Guy Steele and I decided to create a journal for the L&FP community where we could spend more time working with the authors for readability and accessibility. Steele and I spent a lot of time on each submission with revision and rewriting advice.

We had three things in mind when founding the Journal: 1) high quality work deserved a long voice; 2) practical papers deserved an outlet alongside the rising tide of theoretical papers; 3) even the most difficult papers could be made more accessible to a wider audience if authors took the time to learn and practice writing skills. Every issue included an editorial, which was really a short essay, written by Guy Steele and me. The idea was to show that it was possible to make relevant observations and arguments using a relatively informal writing style. Together we wrote eight such short essays. We published six of the original (engineering-centric) Self papers (David Ungar, et al) in 1991.

In 1993 we passed the editor-in-chief batons to others. Later the journal was renamed Higher Order and Symbolic Computation, and the journal became a relatively standard theory publication.

1994–present, Hillside Group and the PLoP conferences: In 1993 I became one of the founders of The Hillside Group, an educational nonprofit aimed at supporting work on design- and software patterns. Because creating pattern languages is really creating a literature, I suggested we should take that seriously by trying to teach and develop patterns writers. I was primary designer of the PLoP conferences: submissions are taken as usual, there is a program committee, but each submission is assigned a *shepherd* whose task is to help the author improve the submission through several rounds of revisions. A program committee member assigned to the paper oversees all the interactions between shepherd and author. After shepherding, the shepherd and PC member make recommendations to the program committee. Accepted authors are invited to a *writers' workshop*, which is a formal review / teaching process borrowed from the creative writing community. After PLoP is over, the authors further revise their works which are then submitted for publication elsewhere.

PLoPs have been held every year since 1994; in 1995 EuroPLoP was founded on the same principles and practices, and in 2001 Hillside Europe was created. In addition there is an annual ChiliPLoP, a bi-annual SugarloafPLoP (South America), an occasional VikingPLoP in Scandinavia, an occasional KoalaPLoP (Australia and New Zealand), and recently an annual AsiaPLoP.

A few years ago the Hillside Group started a Journal called *LNCS Transactions on Pattern Languages of Programming*.

In 2006 we began co-locating PLoP with OOPSLA, and since 2009 PLoP has been co-sponsored by SIGPLAN, giving its authors the opportunity to publish in the ACM Digital Library.

1995–1998, MFA in Creative Writing: In 1995 I decided to take writing much more seriously and earned an MFA in Creative Writing (Poetry).

2002, Onward!: In the early 2000s, SIGPLAN invited me into the OOPSLA fold as a member of the organizing committee. I felt that the standards for reviewing in programming language conferences and the kinds of papers being submitted and accepted made it difficult for researchers to submit early ideas where there were good arguments for the new ideas and perhaps some early results that hinted the ideas just might work. I created a new "track" called *Onward!*. Here is that first call: http://www.oopsla.org/2002/2n_onward.html. The first "proceedings" was not acceptable to SIGPLAN or ACM, so I published them myself here: http://dreamsongs.com/Files/Onward!Proceedings.pdf.

In 2003 I managed to get the Onward! papers published in the OOPSLA Companion. In 2005 as Program Chair for OOPSLA, I was able to convince SIGPLAN to publish the papers in the OOPSLA proceedings. In 2011 I was able to convince SIGPLAN that Onward! should be its own conference and to publish its own proceedings.

Here is the definition of Onward! as written in 2005:

> *Onward! is a separate track that explores or examines new ideas, approaches or paradigms; it includes refereed papers, panels, invited talks, a session called Breakthrough Ideas, a Film Festival, a keynote, and reviewed presentations.*
>
> *Papers describing new paradigms or metaphors in computing, new thinking about objects, new framings of computational problems or systems, and new technologies are presented in the Onward! Track. Such papers don't necessarily advance the state of the art, but aim, instead, to alter or redefine the art by proposing a leap forward—or sideways—regarding computing. Although an Onward! Track paper might not contain a fully worked out theory or implemented system, it must be well thought out, well-written, and compelling in its vision or uniqueness of thinking. Onward! papers are published in the [OOPSLA] Proceedings.*

2005, OOPSLA Essays: In 2005 as Program Chair I added another publication venue to OOPSLA, which later moved to Onward!—***Essays***: some important contributions to our *scientific understanding* of computer science come from deep reflections on technology, science, its applications, and its role in other sciences and human endeavors—presented in a genre not much like scientific and technical computer science papers.

Here is how Essays were described in 2009:

> *An Onward! essay is a thoughtful reflection upon software-related technology. Its goal is to help the reader to share a new insight, engage with an argument, or wrestle with a dilemma.*
>
> *A successful essay is a clear and compelling piece of writing that explores a topic important to the software community. The subject area should be interpreted broadly, including the relationship of software to human endeavors, or its philosophical, sociological, psychological, historical, or anthropological underpinnings. An essay can be an exploration of its topic, its impact, or the circumstances of its creation; it can present a personal view of what is, explore a terrain, or lead the reader in an act of discovery; it can be a philosophical digression or a deep analysis. It can describe a personal journey, perhaps that by which the author reached an understanding of such a topic.*

Essays have been the hardest venue for computer scientists to write successfully. The essay form is foreign to them and so they typically fall back into their technical-paper-writing habits. And what's worse: it's next to impossible to convince program committee members to use any criteria beyond what they use for all program committees. A typical view of Onward! Research papers and Essays: they are ordinary OOPSLA papers with lousy validation.

2005, The Dynamic Languages Symposium: Because there seemed to be no good, major outlet for papers in dynamic programming languages—Lisp, Smalltalk, Python, Ruby, etc—a few others and I started the Dynamic Languages Symposium. In 2010, DLS was added to the list of SIGPLAN conferences.

2005–2010, OOPSLA→Splash Transformation: In 2005 it became clear that attendance at OOPSLA was on the decline, and that the "magic" of OOPSLA was in jeopardy. That magic had to do with bringing together theoreticians, programming language people, software developers, and managers. I put together a study group and we determined that smaller, commercial conferences were eating into our numbers by being better niche conferences than we were a general conference. In 2010 we convinced SIGPLAN to create SPLASH—*Systems, Programming, Languages, and Applications: Software for Humanity*—and put OOPSLA, Onward!, and the Dynamic Languages Symposium under its umbrella.

2007, General Chair OOPSLA: In 2007 we tried to see what it would be like to run OOPSLA as a federated conference, and so we invited quite a few conferences and workshops.

2011, AOSD / AOSA: In 2010 I was asked to help AOSD 2011 with their new venue, Modularity Visions, and with rebranding the conference (this year it was called *MODULARITY: aosd•2012*). We decided to position the venue as requesting submissions that were like good NSF proposals but without the obvious proposalish boilerplate: a well argued proposal for how to think about modularity going forward with perhaps some initial evidence that the approach could work. I was elected Steering Committee Vice Chair.

2009–2011, Steering Committee Chair, OOPSLA and then Splash: Over these years my task was to revive or at least reformulate the conference for long-term health. During this period we tried two approaches to solve the problem of mostly small, safe papers being published at OOPSLA (not Onward!). The first one was a whiteball system in which each (brave, probably senior) program committee member could unilaterally accept a paper, and the second was to use this question as the prime criterion for acceptance: *Is it better for the community to have this paper published than not?*

Whiteballs failed: each of the few papers that were whiteballed was chosen because a committee member believed the submission had been unfairly reviewed. This indicates that there was some belief that reviewing was not entirely fair at that program committee meeting.

The second experiment resulted in a record number (61) and percentage (37%) of papers accepted at OOPSLA. This was cause for debate at the Steering Committee and SIGPLAN Executive Committee levels.

In 2011 I served as Publications Chair for Splash.

Other: I've published five books. I've written a poem a day since March 18, 2000. I am currently working on a hybrid novel / memoir.

Positions

Scope of Papers: We need publication venues (conferences and journals) that publish engineering papers alongside theory and scientific papers. In the "fields" of programming languages and software engineering, engineering work typically precedes scientific work. Would there be Javascript papers at PLDI and OOPSLA if an engineer named Brendan Eich had not created it? Or Java papers if Gosling and his group had not put Java together? These were engineering efforts that created the "nature" our 'scientists' study *today*. L&FP and LaSC were venues that brought engineers and programming language theory people together and they worked fine. What doesn't work are "Experience Reports" and "Practitioner Reports."

Stage of Work and Genre of Papers: We do science by putting ideas on the table and vetting them through a variety of validation techniques. There is scientific value to publishing research proposals (Modularity Visions, Onward! papers)—that is, papers that report ideas, arguments for those ideas, but not yet full validation of them. Moreover, coming to truly understand what we've created through engineering and science takes deliberate rumination. There is scientific value in publishing essays—reflections on what technological and scientific ideas mean and how they are connected. This is the calm aftermath of vigorous scientific progress. We need all these types of publications:

- *Visions and proposals (AOSD Modularity Visions)*
- *Early ideas and arguments for them (Onward! Papers)*
- *Engineering papers well reviewed and structured (like the first **Self** papers by Dave Ungar and colleagues); such papers include case studies and other sorts of material relevant to establishing what the material stuff is for CS&SE scientific study*
- *Regular technical and scientific papers with strong validation, through proofs, statistical analysis, or whatever are taken to be good validation techniques*
- *Essays providing reflection and perspectives (Onward! Essays)*

All publications listed so far must be peer reviewed; this ensures that in the long term, the best researchers will publish such papers because they will get tenure-promoting credit.

Reviewers of New Genres: When scientists "trained" to review scientific papers are asked to review papers of a different genre (Onward! papers, Essays. Modularity Visions), they typically (unconsciously) use a standard roughly akin to this: "a good *<genre>* paper is an ordinary traditional scientific paper for this venue but with lousy or nonexistent validation." Until there is an established literature of nontraditional papers in our scientific publications, we need the following:

- *ways to identify program committee members able to properly judge a nontraditional paper*
- *ways to describe and codify criteria for nontraditional papers*
- *ways for program chairs to enforce criteria for judging nontraditional papers*
- *perhaps ways for authors of rejected papers to appeal decisions*

Shepherding and Other Support: The founders of the design and software patterns community realized that their new genre (patterns and pattern languages) was nonstandard. They established a couple of nonstandard (for computer science) mechanisms to help authors in the new genre succeed: shepherding and writers' workshops.

Shepherding: people accomplished in writing a particular genre are assigned to authors who are less experienced. We need to explore whether such a mechanism should be put in place for nontraditional paper venues until the need passes.

A writers' workshop is a mostly formal process of review led by a moderator or a pair of moderators and consisting of about 6–10 paper authors. The teaching material for the workshop is the set of author papers; the moderators lead each teaching / review session, authors critique (constructively, by design) the papers (authors thus learn critical skills to apply to their own writing), and authors learn to listen to criticism by being required to sit silently while their paper is discussed. We need to explore whether such a mechanism should be used for venues with nontraditional papers. I wrote a book on this process: http://dreamsongs.com/Books.html.

Teaching Writing: We need to up the games of our technical paper authors. For OOPSLA 2010, Martin Rinard, the program chair, wrote the following in the call for papers:

> *There is no page limit on submitted papers. It is, however, the responsibility of the authors to keep the reviewers interested and motivated to read the paper. Reviewers are under no obligation to read all or even a substantial portion of a paper if they do not find the initial part of the paper interesting.*

Several community members were perturbed by this statement, believing it was the responsibility of assigned reviewers to completely and thoroughly review submissions. We need to look into what kinds of recommendations to make to graduate schools for educating our researchers to be at least adequate writers.

Copyright & Ownership: Most publication venues for CS&SE papers require authors to assign copyright to those venues. The venues then generally permit the authors to place the papers on their personal websites and produce updated versions. This permits the venues—most of which are professional societies—to keep the papers available to future researchers regardless of the fate of the authors. Most of these arrangements were designed before two important innovations came along: Creative Commons Licenses and the Web. The Web enables papers to be easily accessible, which was not true in the print-only era; CC licenses are designed to permit the original author to retain the copyright while giving all the rights to the publication venue that they need and like.

However, some venues—like the ACM Digital Library—publish papers from conferences that do not produce final versions, such as PLoP where the publication record consists of the revisions made to papers after passing through the writers' workshop process—these are *works in progress,* which are valuable to the community. For authors of such papers to give up copyright is to essentially cut off further revision and refinement. Another way to put it is that the current publication model supposes final papers.

This gives rise to the following observations and suggestions:

- *the role of an academic publisher, particularly a professional society, is to retain the record of peer-reviewed or otherwise editorially selected scientific, technical, or engineering papers; call such a place a "**peer-review repository**"*
- *each paper in a peer-review repository should have a link to the peer-review group that reviewed and accepted it*
- *copies of papers from a peer-review repository should have both a visible stamp of authenticity as well as an effective (if there be one) non-counterfeitable, invisible indicator of authenticity*
- *authors retain copyright but licenses make it clear that a peer-review repository holds the "official," peer-reviewed version; versions held by the author should also bear a mark pointing to the peer-review-repository version*
- *peer-review repositories must keep detailed and accurate bibliometric records of citation counts for both individual papers and for the conferences, journals, and other venues that certify them (some professional societies, such as ACM, do a terrible job on conference-wide bibliometrics—see Note 1)*

Living Documents: These days, research "documents" are not embodied only as ink on paper, but also as digital text files, source code, data sets, videos, and other media. Despite the best effort of peer-review panels, mistakes are made. Therefore, corrections to archived research documents should be permitted, as should adding new data, new analyses, and related materials. These errata, updates, and other addenda should be part of a living document centered on a peer-reviewed paper or set of related papers, and peer-review repositories are the right places for such living documents. It is important that all such updates leave the original paper untouched—only additions should be permitted.

Note 1: Consider OOPSLA, which is a SIGPLAN conference. Visit this web page:

http://dl.acm.org/event.cfm?id=RE181

You will read that in the 26 years OOPSLA / Splash has been running, there have been 2,889 papers / articles published. Pretty impressive. However, if you do the division, you'll find that these numbers imply that there were over 111 papers accepted each year. This is pretty remarkable because in 2011 OOPSLA set a record—and created a huge controversy—by accepting 61 papers (a record for most papers, not fewest). The effect is that OOPSLA's reputation for some is as a low-impact conference, with only 7.42 citations per paper; compare that to another SIGPLAN conference, PLDI, which accepts the same types of papers using the same program-committee pool—PLDI has 30.92 citations per paper. If we assume the typical OOPSLA year is like the record year, that would mean make it 1,586 papers and 15.31 citations per paper; if we assume about 33 papers per year over the whole period—which I believe is the likely number—then there are 858 papers and nearly 25 citations per paper, which puts OOPSLA in the same category as PLDI—this is how OOPSLA is perceived in the community **today**.

The error likely occurs because OOPSLA also publishes a companion to the proceedings, which contains workshop, panel, and tutorial descriptions; workshop papers; and non-peer-reviewed materials from co-located events. The ACM Digital Library apparently lumps the companion materials in with papers from the peer-reviewed proceedings.