

EVENTS AND SIGHTINGS

Report on HOPL IV—ACM SIGPLAN History of Programming Languages Conference

Mark Priestley, *The National Museum of Computing, Bletchley Park, U.K.*

Tomas Petricek, *University of Kent, U.K.*

HOPL IV is the latest in the ACM's occasional series of conferences on the history of programming languages. Originally scheduled for London in 2020, the conference was postponed because of the Covid-19 pandemic, and took place online on June 20–22, 2021, in association with PLDI 2021.

HOPL conferences take place at roughly 14-year intervals. The first HOPL, organized and chaired in 1978 by Jean Sammet, contained 13 invited papers surveying the early development of influential programming languages that had been in use for at least 10 years before the date of the conference. The second conference, in 1993, accepted both invited and contributed papers, and broadened the scope to include papers on language evolution, the history of language features, and classes of languages. HOPL III in 2007 adopted a similar model, enlarging the scope slightly to cover influential research languages that may not themselves have been widely used. Following in this tradition, the HOPL IV program chairs, Guy Steele and Richard Gabriel, issued a call for papers in the autumn of 2017 and also sent out 62 “letters of encouragement” soliciting specific contributions. A particular effort was made to obtain papers on significant languages that had not been included in previous HOPLs, such as Objective-C and the pioneering functional language ML. Unfortunately, it proved impossible to obtain a paper on Java, one of the most important languages of the last couple of decades. This process led to 22 submissions, of which 19 were eventually accepted.

While many languages were making their first appearance at HOPL IV, “evolution” papers covered

the continued development of APL (first appearing in HOPL I), C++ (HOPLs II and III), and Smalltalk (HOPL II). Papers on hygienic macro technology and on coarrays and SPMD parallelism in Fortran covered specific language features. Three papers covered members of the family of languages inspired by C (HOPL II): Bjarne Stroustrup discussed the ongoing history of C++, focusing on revisions of the ISO C++ standard since 2006; Brad Cox and his coauthors described the early development of Objective-C, which combined the C language with a Smalltalk-inspired approach to object-orientation;¹ and Walter Bright presented the history of the D language, an attempt to improve on C++. The broader scope of HOPL IV also accommodated a paper on the multiparadigm Oz language, whose layered structure makes it a popular and influential language for teaching programming concepts and paradigms.

The program committee believed that every submission contained valuable historical material, and aimed to convert as many as possible into publishable papers. Recognizing that many authors were language designers with no experience of writing extended historical narratives, the program chairs assigned each paper a PC member as a “shepherd.” The shepherd guided the authors through at least two major revisions of the original draft, offering detailed comments and advice on style, structure, and content. This intense collaboration placed heavy demands on the shepherds, but reaped rewards, with several authors commenting on the improvements that the process had made to their initial drafts.

HOPL conferences have always maintained links with the history of computing community, in the form of a “consulting historian.” Following Henry Tropp (HOPL I), and Michel Mahoney (HOPLs II and III), Mark Priestley filled this role for HOPL IV. The consulting historians' role has been to offer guidance to the program committee on basic practices of historical research and writing. Starting with the pioneering

work of Sammet and others in the 1960s, the programming language design community very early adopted an explicit and historically responsible interest in its own history, and the HOPL IV program committee continued to uphold these high standards. Potential authors were provided with detailed lists of questions to help structure their historical research and were recommended to read Mahoney's short paper "What makes history?," originally prepared for HOPL II.²

The 19 HOPL IV papers have been available in the ACM Digital Library since June 2020, the original planned date of the conference.³ They are rather different from much current work in the history of computing, and some may strike historians as being long and technical. This is a feature, however, not a bug. HOPL papers reflect the historical interests and priorities of the community of programming language designers and aim to capture historical "raw material" from language designers, be it in the form of narrative, otherwise unavailable documents, or reminiscence, and present it in a coherent and well-sourced form. The HOPL process encourages the collection and indexing of primary source material that might otherwise be lost. In many cases, this material is then presented in the form of a first-person narrative that provides much of the liveliness and quirky detail found in good oral histories, shaped into greater coherence by the shepherding process.

The online conference platform allowed prerecorded talks to be viewed, and captured video streams of all the conference sessions, including discussion periods where presenters addressed questions submitted by the online audience. A number of presenters took advantage of the online format to include a wider range of speakers than would have been feasible at a physical conference, or to explore lively conversational and multimedia styles of presentation.⁴

The conference included two keynotes that suggested a range of perspectives on programming languages in which the more detailed papers could be viewed. In her talk "Myths and mythconceptions," Mary Shaw set programming languages in a broad context of use, highlighting and critiquing a number of "myths," or characteristic ways in which computer scientists think about programming languages and software.

Questioning the idea that programs are mainly written by highly skilled professional programmers, Shaw highlighted the importance of "vernacular programmers," and a number of presentations discussed languages aimed at this constituency. Cynthia Solomon led a fascinating and wide-ranging presentation describing Logo in the context of a culture for teaching mathematical thinking (and thinking about

thinking) to children. Other presentations, such as those by Cleve Moler on the history of MATLAB, John Chambers on "S, R and Data Science," and Jeffrey Kodosky on the graphical language LabVIEW, covered languages widely used by mathematicians, scientists, statisticians, and engineers for numerical computing, data analytics, and instrument control.

Shaw also identified as a myth the idea that programs are closed systems defined solely by their code, pointing out that much modern software is a coalition of many types of components and supporting tools. A number of presentations discussed programming languages that exist as components in a broader ecosystem, notably JavaScript, whose history, presented by Allen Wirfs-Brock and Brendan Eich, is inseparably linked to the history of the World Wide Web and the "browser wars" of the 1990s. Other languages that are inevitably viewed as parts of broader technological coalitions are those targeting widely adopted multilanguage platforms, namely the Java Virtual Machine and Microsoft's .NET. In this category, HOPL IV included presentations on Clojure by Rich Hickey, F# by Don Syme, and Groovy by Paul King. Programming languages embedded in broader contexts of use were discussed in presentations by Peter Flake and colleagues on the history of the hardware description language Verilog HDL, and Stefan Monnier and Michael Sperber on the history of the Lisp dialect embedded in the Emacs editor.

The second keynote, "programmingLanguage as Language," was presented by James Noble and Robert Biddle. It suggested ways in which programming languages can be viewed as "languages," using the perspectives of formal and machine languages as well as human language. The theme that language provides a convenient notation not just for writing, but also for thinking, was picked up in a talk by Roger K.W. Hui and Morten J. Kromberg on "APL since 1978," about a programming language widely known for its terse syntax. Noble and Biddle also considered language as a communication mechanism, looking at reflections of ideas of conversational and discourse structure in programming. Daniel Ingalls' presentation on the development of Smalltalk gave a very clear illustration of a language intended as an intrinsic part of a broader interactive programming environment (a theme also present in the case of Logo) and was delightfully illustrated with live examples of reincarnated systems from Smalltalk-72 to Squeak.⁵

Taken as a whole, the HOPL IV papers and presentations are testimony to the continuing vitality of the field of programming language design. In addition to their intrinsic interest, reading them in the

perspectives offered by the keynote talks provides insight into the importance of programming in a very wide range of contexts. Program chairs Guy Steele and Richard Gabriel are to be congratulated for their success in guiding a large and extremely hard-working program committee through a long and complex process (and a global pandemic) and delivering a conference that upheld the high standards of previous HOPLs and provided a fascinating snapshot of the state of programming languages in the early 21st century.

REFERENCES/ENDNOTES

1. The conference noted with sadness the passing of Brad Cox, the originator of Objective-C, and acknowledged with gratitude the work of Hansen Hsu on bringing this paper to completion.
2. Detailed information about the editorial process can be found at: <https://hopl4.sigplan.org/track/hopl-4-papers>.
3. They form the special issue "HOPL" in *Proc. ACM Program. Lang.*, vol. 4. [Online]. Available: <https://dl.acm.org/toc/pacmpl/2020/4/HOPL>.
4. Recorded talks (including keynotes) can be viewed on the PLDI 2021 website at https://www.pldi21.org/track_hopl.html, but video streams and question and answer sessions are not available. The HOPL IV chairs expect that all recordings will eventually be available at <https://acm.dl.org>.
5. These are available at the online "Smalltalk Zoo" hosted by the Computer History Museum. For an introduction, see: <https://computerhistory.org/blog/introducing-the-smalltalk-zoo-48-years-of-smalltalk-history-at-chm>.

MARK PRIESTLEY is currently a senior research fellow at the U.K.'s National Museum of Computing, Bletchley, U.K. He works on the history and philosophy of computing, with a focus on programming in the 1940s and 1950s. Contact him at m.priestley@gmail.com.

TOMAS PETRICEK is currently a Lecturer in computer science at the University of Kent, Canterbury, U.K. He is interested in finding easier and more accessible ways of thinking about programming. To do so, he combines technical work on programming languages and tools with research into history and philosophy of science. Contact him at tomas@tomasp.net.

SHARE AND MANAGE YOUR RESEARCH DATA

IEEE DataPort is an accessible online platform that enables researchers to easily share, access, and manage datasets in one trusted location. The platform accepts all types of datasets, up to 2TB, and dataset uploads are currently free of charge.

 Open Access Options	 Generates Citations	 2 TB of Cloud Storage	 Links to Manuscripts
 Reproducible Research	 ORCID Integration	 Hosts Data Competitions	 DOI Provided

IEEEDataPort
UPLOAD DATASETS AT IEEE-DATAPORT.ORG