



会議レポート

PLDI 2016 報告

PLDI 2016 とその共催イベント

プログラミング言語分野のトップ国際会議として知られる, the 37th annual ACM SIGPLAN conference on Programming Language Design and Implementation (PLDI 2016) が 2016 年 6 月 15 ~ 17 日にかけて開催された。本稿では, PLDI 2016 に参加した筆者によって, その概要を, 筆者の主観的意見および印象とともに紹介する。

PLDI は, 冠する名の通り, プログラミング言語の設計と実装に関する会議であり, 多くのトップ国際会議と同様に, 関連する話題の会議が共催される。図-1 に, PLDI 2016 の代表的な共催イベントをまとめる。LCTES は, 組込みシステムのためのプログラミング言語に関する話題を取り扱う国際会議であり, ISMM は, メモリ管理に関する話題を取り扱う国際会議である。どちらのスコープも, おおよそ PLDI のスコープに包含されるため, より小さく専門的な会議といえる。Tutorials は, PLDI 本流の研究話題である言語処理系, 解析器, デバッガ等における, 最新ツールを紹介する。ツールはすでに論文として出版されているものがほとんどである。本稿では, PLDI 2016 本体のみに焦点を当てる。

研究論文

PLDI 2016 には, 過去最多の 306 本の投稿があり, そのうち 48 本が採択された。採択率は 16% であり, トップ国際会議の名に恥じない水準である。本章では, PLDI 2016 の採択論文とその査読方法を紹介する。

査読

PLDI 2016 では, 2 段階の完全な double-blind 査読が採用された。ここでの完全とは, 査読プロセスのいかなる段階でも, 著者情報が査読者側に公開されず, 最終的にも採択論文の著者情報しか公開されないという意味である。具体的には, 第 1 ラウンドでは各論文に 3 人の査読者が割り当てられ, その結果を持って査読論文を絞り込む。ふるい落とされた論文は, この時点で通知される。第 2 段階では, 2 名以上の査読者を追加して査読し,

6/13	6/14	6/15	6/16	6/17
LCTES		PLDI		
	ISMM			
Tutorials				

図-1 PLDI 2016 の日程概観

査読者の議論の後, プログラム委員会 (PC) が採択を決める。筆者の知る限り, 完全な double-blind を採用するのは, 過去の PLDI にはない試みである。この結果として, 「査読論文の著者が知っている人だと思うか」という質問に, 68% の査読者が No と答えることとなった。ちなみに, ほかのプログラミング言語分野 (ACM SIGPLAN) のトップ国際会議である POPL 2016, ICFP 2016, OOPSLA 2016 では, 過去の PLDI と同様に, 軽量の double-blind, すなわち著者情報を最初の査読では非公開にして, それを受けた PC 議論においては公開する方式が採用されている。完全な double-blind を先駆けて採用した PLDI 2016 は, 査読のバイアスを最小化する意識が一層高いことがうかがえた。

完全な double-blind に加えて, PLDI 2016 では, 査読の品質向上のために, 2 つの新しい措置が導入された。1 つは, word count である。短い査読レポートは論文著者のためにならないので, 査読レポートが 500 単語を超えるように, 入力フォームが注意喚起していたようである。実際にその効果はあり, PLDI 2015 と比べて全体的に査読レポートの長さは, おおよそ倍になっていた。もう 1 つは, 査読フォームに明確な形式を用意したことである。査読者の専門性 (expertise) を説明し, さらに査読論文個別の専門性も説明する欄を用意した。そして, 査読論文を擁護 (champion) する説明を与える欄も用意した。これは, 投稿論文と査読側との専門性のギャップを埋めることと, PC 議論の円滑化を目的としたものである。実際, PC chair's report において, 本当に論文が理解できる人に査読を割り当てたことが強調されていた。専門性を基準に博士課程学生に査読を依頼することもあったとのことである。プログラミング言語に関する広いスコープを, 最高の水準で扱い続けようとする, トップ国際会議としての誇りを感じた。

詳細が気になる読者は, Proceedings の Front matter および, PC chair による文書レポジトリ^{☆1}を参照されたし。

研究話題

採択論文の研究領域の公式リストは存在しない。このことについて, PC chair は, “Area is PLDI!” と説明し, 会場から笑いを取っていた。この発言は, 単なる冗談ではなく, プログラミング言語に関する多様な話題を扱っ

☆1 <https://github.com/emeryberger/PLDI-2016/>

ているという、ある種の自負の表れであろうと思う。しかし、公式リストがないとはいえ、プログラムのセッション名から、研究話題のおよその傾向を見て取ることはできる。

2セッション分以上用意されていたのは、Down to the Metal (6件)、Verification / Verifying Systems (8件)、Types (6件)、Synthesis (5件)、そしてParallelism (5件)である。ここで、Verification (5件)とVerifying Systems (3件)は異なるセッションであり、前者は検証技法に、後者は大規模なシステム(OSやコンパイラ)の検証に重点があるという違いを見て取れるが、研究話題の分類としては1つにまとめた。少々ひねった名前のDown to the Metalは、アーキテクチャやバイナリなどの低水準の話題を意味する。その内訳を、筆者の主観で分類すると、意味論1件、デバッグ2件、型推論1件、プログラム合成1件、静的解析1件であり、低水準という共通点はあるものの多様な話題を含んでいる。1本の論文が複数の話題を伴うことはよくあるので、上述のセッション内件数は、全体の研究話題の分布を直接意味しないのだが、およそ反映した結果となっている。

Distinguished Papers

近年のPLDIでは、採択論文の最大10%分だけ、Distinguished Papersが選ばれる。PLDI 2016では、前述の主要セッションであるDown to the Metal, Types, Parallelismからそれぞれ1件、Memory Managementセッションから1件、合計4件選出された。

ここでは、Distinguished Papersの中で、筆者が最も注目し評価した論文であるMemarianらの"Into the Depths of C: Elaborating the De Facto Standards"を簡単に紹介する。これは、端的に言えばCの形式的意味論を定義した論文である。ISO C11標準から、彼らが設計したコア計算であるCerberusへの翻訳を与えている。ここまでだと凡庸な話に見えるが、重要なのは、彼らはISO標準以外の意味論も考慮に入れたことである。

タイトルの通り、de facto standardsを扱っている。これが複数形になっていることが、Cコンパイラの現実として重大な問題である。ISO標準は、自然言語で定義され、曖昧もしくは不明瞭な点が多くあるため、処理系依存性は不可避である。そして、ISO標準は抽象的でもある。たとえば、物理アドレス上で隣接する局所変数へのポインタ演算によるアクセスは仕様違反(未定義動作)である。この抽象性や処理系依存部分を利用して、Cコンパイラは最適化を行うので、コンパイル後の物理メモリモデルを、ISO標準より強い仕様として単純に採用してはならない。ISO標準全体が、現実的に妥当というわけでもない。たとえば、異なる配列上のポインタ大小比較は、ISO仕様では未定義動作である。しかし、現実にはそういうコードは大量にあるので、現実のCコンパ

イラは、そこに妥当な意味を与えなければならない。このように、ISO標準には、さまざまな側面で際どい部分が多くある。彼らはそれを85項目に分けて、de facto standardsを調査し、その上でCerberusを設計した。そして彼らは、実験的にGCCやClangの複数のバージョンとさまざまなフラグについて動作検証を行い、異なるde facto standardsの下でも適切にCerberusに翻訳できることを実証した。この研究には正気を疑うレベルのコストがかかっているというのが、筆者の率直な感想である。7人による共著であるのも十分頷ける。この論文の謝辞から、英国政府や米国政府などから、複数の資金調達を行っていることが分かり、この研究の仕事の大きさを物語っている。

基調講演

PLDI 2016では、Benjamin Zorn (Microsoft Research)とLuiz André Barroso (Google)による、2件の基調講演があった。どちらの話も、背景や導入には、ビッグデータとディープラーニングが出てきた。しかし、(特に前者の講演では)それらがバズワードとなっていることを分かった上で、あえて話のネタにしている雰囲気であった。具体的な内容としては、前者は、IoT時代におけるソフトウェア危機についての話、後者はデータセンタにおけるフラッシュデバイスの活用についての話であり、話者の専門性(出版論文)と所属を反映していた。

筆者の印象に残ったのは、Zornの講演である。とても上手で、聴衆を飽きさせない面白いプレゼンであった。最初に公衆衛生(public health)、財政の安定(financial stability)、そして信頼と保障(trust & security)を改善する話をする切り出して、一体どうプログラミング言語に繋がるのかと思われた。話を聞いていると、公衆衛生は、ジカ熱の予防のために蚊を収集・解析する自律学習システムの話であり、その学習の部分がプログラミング言語における強力なツールである定理証明系に結びついた。財政の安定は、会計処理に広く使われるスプレッドシートのバグの話であり、非プログラマのためのプログラム合成^{☆2}に結びついた。そして、信頼と保障は、製品品質保証に関する話であり、プログラム検証に結びついた。このように、講演内容は多岐に渡ったが、最後に、信頼と保障の話に続く形で、あらゆる製品にソフトウェアが載るIoT時代のソフトウェア危機を示唆した。Hawkingなどの科学者が、人類に対するAIの脅威に警鐘を鳴らしているが、AIはソフトウェアの一部でしかない。(より一般に)誤ったソフトウェアの脅威を考えなければならない。このとき、ソフトウェアが何をしているのかをきちんと把握することこそが真に必要なこと

^{☆2} 関連記事: 森畑明昌: プログラミングするプログラム—自動プログラム作成最前線—, 情報処理, Vol.57, No.6, pp.544-549 (2016).

であり、そしてそれはプログラミング言語の問題であると結んでいた。本講演から、Zorn が、明確な先見の明をもって、非常に大きな立場でプログラミング言語研究に取り組んでいることが伝わり、率直に凄い人だと感心した。

全体の印象

PLDI 2016 への筆者の独断と偏見に満ちた印象を述べて、本稿の結びとする。

まず、研究話題の変化について、近年の PLDI に見られたプログラム合成の流行は、一段落したように思える。いまだに合成 (synthesis) という単語は論文タイトルに多く見かけるが、合成できることではなく、合成対象に焦点が移り始めたという印象である。また、論文件数の多い検証と合成のセッションから、Distinguished Papers が選出されなかったことは、それらが当たり前になったことで、真新しさが減ったこと示唆しているように思えた。なお、筆者視点での PLDI の特色と近年の傾向は、PLDI 2015 報告^{☆3}にて述べられている。

^{☆3} http://doi.org/10.11309/jssst.33.1_15

次に、研究の傾向について、全体として、近年の PLDI と比べて、手堅い印象を受けた。アイデアとして特段面白くはなくても、しっかり実装してあることを評価するという査読方針だったのではないかという印象を持った。ただし、PLDI のスコープの観点から疑問符がつく採択論文もあった。高い専門家性を持った査読者が、論文を擁護する査読形式を取った副作用かもしれない。

最後に、日本 (人) の存在感について、日本からの PLDI 2016 への出席者は、合計 5 人 (3 大学から教員 4 人、留学生 1 人) であった。日本のプログラミング言語研究者数を踏まえると、少なすぎると言っても過言ではない。日本からの採択論文がなかったことは、トップ国際会議の競争の激しさから仕方がない面はある。しかし、プログラム委員会 (35 人)、外部プログラム委員会 (24 人)、外部査読委員会 (71 人) の総勢 129 人の中に、日本人 (名の人) や日本国内の所属の人が誰もいない。これは、「PLDI という Area」(PC chair の発言から借用) での日本の弱さを如実に表している。日本の PLDI ファンの 1 人として、この惨状に悲愴感すら覚えるが、他人事で済まされないという思いが強くなった。

(佐藤重幸 / 高知工科大学情報学群)

