Peter J. Landin (1930–2009)

Olivier Danvy

Published online: 13 January 2010
© Springer Science+Business Media, LLC 2010

Abstract This note is a prelude to a forthcoming special issue of HOSC dedicated to Peter Landin’s memory.

In those days [the 1960’s],
many successful projects started out
as graffitis on a beer mat
in a very, very smoky pub.
Peter J. Landin, 2004

One of the founding fathers of everything lambda in programming languages passed away in June 2009: Peter J. Landin.

Peter Landin spent the last years of his life as Professor Emeritus at Queen Mary, where his colleagues included Edmund Robinson and Peter O’Hearn. Over the last decade, he served in the advisory board of HOSC and thus received a complimentary copy of each issue. HOSC published two of his articles: a tribute to Christopher Strachey [21] in 2000 and a reprint of his 1965 technical note “A generalization of jumps and labels” [20] in 1998, for which Hayo Thielecke wrote an introduction [30]. In the editorial of our 1998 issue [10], Carolyn Talcott and I presented this reprint as follows:

This paper describes a real conceptual discovery, namely the idea to make control facilities first-class entities in a programming language, through the “J operator.” Its exposition is typical of the simplicity, directness, clarity and honesty of Landin’s writing that makes his articles such a pleasure to read.

This spring, for the last time, I sent him a copy of a scientific compliment: a joint re-visititation with Ken Shan and Ian Zerny of his direct-style embedding of Algol 60 into applicative expressions with the J operator [15]. There, we retarget his embedding to the Rhino implementation of JavaScript with continuation objects. Indeed, whereas call/cc captures
the current continuation, the J operator captures the continuation of the caller of the current method [7]. This feature fitted Landin’s embedding then and it fits a JavaScript implementation with a local stack for each method now [3]. Playfully, the title of our re-visititation is thus “J is for JavaScript” [11].

In 2004, I paid Peter Landin a visit at the occasion of Josh Berdine’s PhD defense [2] and found him in his office, patiently helping an undergraduate student. In turn, I patiently waited for him to be done with the student before presenting him my rational deconstruction of his SECD machine [5]. I then showed him how the SECD machine could be put into defunctionalized form [9, 25] and could then be refunctonalized [8] into a continuation-passing evaluator à la Lockwood Morris [22]. I thus enthusiastically concluded how much the SECD machine made sense, and that even though he might not have discovered continuation-passing style (see Appendix), defunctionalization and refunctonalization provided a concrete argument why his name should be added to the list of the discoverers of continuations [26]. Throughout, he was as patient with me as with the undergraduate student, and in the end he smiled, his eyes sparkled amusedly, and then he made some incredibly modest comments to the effect that he had been lucky.

Peter Landin was indeed so modest that in 1998, he did not attend the MFPS XIV session held in his honor at Queen Mary, 1 eliciting Dana Scott’s quip as to whether Peter Landin was the Bourbaki of Computer Science. He did, however, get to read hardcopies of the slides displayed at his session.

I initially got in touch with Peter Landin in 1996 by e-mail and by phone and we met for the first time in January 1997 in Paris, at the occasion of the Second ACM SIGPLAN Workshop on Continuations [4], which I was chairing and where he gave a keynote speech. For the proceedings, he wrote the masterfully idiosyncratic “Histories of Discoveries of Continuations: Belles-Lettres with Equivocal Tenses” [19].

After his keynote speech at CW’97, he handed out copies of some of his old research reports [16, 17]. There naturally was a stampede, and to Olin Shivers who asked his copies to be autographed he said “I am not the Beatles,” and then signed them.

When introducing him before his keynote speech, I pointed out that independently of all his accomplishments, he was a rare breed of computer scientist with a control operator as his middle name (which is “John” and is abbreviated “J” in his publications). He flashed a look at me that to this day makes me wonder whether it was such a good idea to mention this coincidence at all.

My favorite moment with Peter Landin occurred when we met: he was arriving from London to attend CW’97, I picked him up at the train station, and together with John Reynolds and Andrzej Filinski, we sat at the terrace of a French café. I took the opportunity of a pause in the conversation to venture the question as to whether in their mind, the evaluation order of the meta-language of denotational semantics was call by value or call by name. Peter and John immediately, and simultaneously, answered “call by value of course” (for Peter) and “call by name of course” (for John). For a second of eternity, they looked at each other. Then it was like they were mentally telling each other “let’s not have this discussion again” and the universe resumed its course. The rest of the evening was warm and pleasant, the following day was as wonderful as each continuation workshop somehow manages to be, and eventually I took him back to the train station.

What happened before is history: his impression that computer science was turning “too theoretical” for him, his quiet move away from the programming-language limelight, and

1 http://www.dcs.qmul.ac.uk/~edmundr-mfps/

2 Including “So these continuations have continuations.” which beautifully anticipates the CPS hierarchy [6].