Peter Landin: a computer scientist who inspired a generation, 5th June 1930 - 3rd June 2009

Richard Bornat

DOI 10.1007/s00165-009-0122-y

Formal Aspects of Computing (2009) 21: 393–395

Formal Aspects of Computing

Peter Landin: a computer scientist who inspired a generation, 5th June 1930 - 3rd June 2009

Richard Bornat

E-mail: r.bornat@mdx.ac.uk

Peter Landin, who has died recently of prostate cancer, was an outstanding academic computer scientist who set the direction of the subject for his generation. In an astonishing burst of creativity in the 1950s and 1960s he showed that programs could be defined in terms of mathematical functions, how to translate them into functional expressions in the lambda calculus, the creation of Alonso Church in 1936, and how to calculate their meaning with an abstract mathematical machine, prefiguring many of the advances of the next couple of decades. Although later he turned away from computer science towards gay politics, he remained an original and extreme thinker, a complex and intriguing character.

An only child born in Sheffield, son of an accountant father disabled in WW1, he was educated at King Edward's Grammar School. During National Service he amused himself by trying to drill a squad to imitate addition in binary arithmetic. At Clare College Cambridge he completed the mathematics degree in two years, then attempted the very difficult part 3, but came away with only a 3rd class degree. The emerging computing industry was a haven for mavericks, though, and he took a job at English Electric, where he cooperated in the development of EE's Deuce from the National Physical Laboratory's early Pilot Ace.

His ideas had already turned to the problems of relating the mathematics of logic to the ad-hoc languages invented by programmers. He was involved in the discussions which led to the development of Algol 60, a seminal programming language of the time, though he failed to convince the committee that it should be defined in terms of substitution as in the lambda calculus. His interests drew him into Mervyn Pragnell's discussion group. Mervyn used to prowl bookshops spotting people who were buying books on mathematical logic and invite them to his group which read and discussed the books in a kind of perpetual cycle. Despite the depth of his understanding compared to the rest of the group, Peter was always kind and helpful to novices and stragglers, and even programmers could join and learn.

In 1960 he married Hanne and he was taken on as a research assistant by Christopher Strachey, then an independent consultant and later Professor at Oxford, to write a compiler to translate 'autocode', an early programming language, into the machine language of Ferranti's new Orion machine. He envisaged the compiler as an automatic product of the semantics of the autocode, matching its forms to semantic representations of the instructions of the machine, and generating LISP expressions that could be executed, but his attempt was never finished, and it was a long time before such a radical approach became feasible. He was held in awe by the elite

Correspondence and offprint requests to: R. Bornat, Prof. of Computer Programming, School of Engineering and Information Sciences, Middlesex University, The Burroughs, London NW4 4BT, UK. E-mail: r.bornat@mdx.ac.uk

This is not a biography: it is an attempt at a public statement which will tell people how important Peter was and how much fun he was to be with. I have a lot of biographical clues and one day perhaps somebody will attempt one, preferably before his contemporaries are all dead.

team who were writing the 'supervisor' (operating system) of the machine: he would appear before them, dressed in a long overcoat, and scribble expressions in the lambda calculus, rather small, in the middle of a blank piece of paper, trying to influence them to make their design fit the needs of his compiler.

Even though he never wrote his compiler, his time with Strachey was not wasted. He clarified his ideas about program semantics and published "The mechanical evaluation of expressions" which showed how to translate programs into the lambda calculus and defined the SECD machine, an operational definition of the meaning of a small language which he hoped might form the basis of the design of future computers. In particular he showed how functions and procedures in languages like Algol, which introduced the now universal notation of nested scopes, could be given a concrete meaning in an implementation. Every compiler writer since has had cause to be grateful to him, and modern programming languages such as Javascript, which underpins the World Wide Web, all use functional values heavily and have implementations based on his definition of a 'closure'.

Even at that stage computing was not his entire life. All of his life he was a radical in politics, a regular on demonstrations, and he was arrested at one of the Committee of 100 demonstrations in Trafalgar Square. He was an accomplished musician, and he and Strachey would end technical seminars by playing piano duets. But characteristically he was interested in understanding rather than performance: each new piece was a challenge to be surmounted then surpassed.

In 1964 he was 'brain drained' to the USA and moved with his wife and two small children to New York to work for Univac, then a major computer manufacturer. No expense was spared: though they'd been put in a hotel he asked for a house with a garden, which is what he'd had in London, and they were given an approximation—half a house—in Greenwich Village. He published "A correspondence between Algol 60 and Church's Lambda notation" which drew out in greater detail and with greater fidelity the correspondences he'd shown already, which he later extended to the problematic notions of labels and jumps. In "The next 700 programming languages" he gave a witty account of how *all* programming languages of the time were just sugared versions of the lambda calculus, and he defined ISWIM (If You See What I Mean), an invented but unimplemented programming language which could be used to explain his ideas to those deterred by the austerity of mathematical notation. Like many of his advances, ISWIM was presented casually, but it was much more than a presentational device: it was the first programming language to be given a meaning independent of a particular computing machine, being defined by simple rules to translate it into the lambda calculus.

But he hated corporate work in a skyscraper and in 1966 he moved to MIT. It was not much better there. He was used to sharing his ideas with his colleagues, in Mervyn's group and in loud pub discussions, but in Cambridge Mass. people kept their ideas to themselves. They would not listen to him very much: in particular they did not take to his 'closure' idea, and as a result LISP, the first language to deal with list processing and functional values, got functions wrong in all its implementations for a decade and a half. Peter remarked that "the LISP they understood and the LISP I understood had little to do with one another".

In 1967 he was tempted back to London to take a chair at Queen Mary College, where he remained for the rest of his academic career. He involved himself into teaching and research but he did not publish substantially again though there were a couple of papers on universal algebra and programming semantics, one with Rod Burstall, who he'd met originally in Mervyn's group, and who became a professor at Edinburgh. He worked with John Reynolds, who was his research assistant for a year and later became professor at CMU and a world-leading programming language theorist, and with Hans Bekic of the VDL/VDM group which was working on mathematical definitions of programming languages. He was always trying to explain his ideas about the connections between computing and fundamental mathematical logic to his colleagues and to students. His work continued to have influence: ISWIM profoundly affected the design and definition of ML, and through that even languages such as Java which underpin the World Wide Web.

But increasingly he seemed to be losing interest in computer science. Always bisexual, he threw himself into the politics of the Gay Liberation Front in the early 1970s. He drifted apart from his wife, and they separated amicably in 1973. For the rest of his life he pursued gay politics, and his house in Rona Road became a famous gay commune. He was always more of a facilitator than an activist, though he was once arrested on a GLF demonstration. His talent for inviting people to his dinner parties that he thought should meet hatched many plots. He was always a radical: AIDS the musical, with its chorus of WPCs in rubber gloves, was conceived as a protest against those he felt were not pressing the gay cause as hard as he thought they should and his phrase "an iron fist in a rubber glove" a statement against police elements who played up the 'threat' of interactions with gay people. It was at one of his dinner parties that those who reinvigorated Gay Pride marches in the mid 80s met, just in time for the battle over clause 28.

Never a joiner, always a critic, always original and extreme, annoying but lovable, he kept his academic work to an acceptable minimum but never stopped thinking and writing. Towards the end he became convinced that computing had perhaps been a bad idea, giving support to profit-taking corporate interests and a surveillance state, and that perhaps he'd wasted his energies in promoting it. But he never completely gave up on those who were studying the subject, remaining a frequent visitor to British Computer Society meetings in London, even though he had little sympathy for formal reasoning and was opposed even to variable typing.

He is survived by his wife Hanne, his children Daniel and Louise and by his ideas which remain central to the study of the subject, although computing has outgrown the simply sequential languages of his youth, and are constantly revisited by new researchers wanting to understand how to build on the foundations he laid.

Published online 13 August 2009