

# The Making of Algol 68

C. H. A. Koster  
Informatics department  
University of Nijmegen  
The Netherlands

*Speak, memory ...*

In September 1967, Aad van Wijngaarden, the director of the Mathematical Centre in Amsterdam, asked me to join the ongoing project to define a successor to Algol 60.

I had just finished 18 months in the Dutch Army, and was looking forward to the resumption of a carefree life, working on Natural Language Processing. Van Wijngaarden's eye fell on me for a number of reasons: I was steeped in the spirit and implementation of Algol, well-versed in two-level grammars and I knew everything there was to know about operating systems and I/O, having just completed a vast Fortran program on an IBM 7094 under IBSYS that made use of all its 16 tape drives.

In this way I became one of the Authors of Algol 68, and a participant in one of the formative events of our profession — the making of Algol 68. In this note I will try to give you an eyewitness account. I am not a historian, and have kept very few notes from this period, but the Informal Minutes of the Tirrenia [7] and North Berwick [8] meetings bring back many memories.

Let me describe to you four parties, the Editor, the Authors, the Committee and the Computing Community, taking part in a cosmic struggle for Truth. The period is November 1967 to December 1968.

The persistent Capitalization of certain important substantives in this text is entirely in keeping with the IFIP-style of those times.

## 1 The Computing Community

The design of Algol 68 was firmly rooted in the Computing Community, a contemporary term for the small but growing international community of computer professionals and scientists, whose eyes had been opened by Algol 60, and who were giving shape to the Science and practice of Informatics.

In those days, proceedings of conferences came with a record of the discussion following each presentation. It is fascinating to see the great of that period learnedly discussing, and taking standpoints that foreshadow the various paradigms that over the last decades have come to split the community.

But the most interesting discussions could be found in the Algol Bulletin, edited by Fraser Duncan. Many of the criticisms of Algol 60 contributed to the aims of Algol 68, as described in chapter 0 of [MR101]. Many features found in Algol 68 were first proposed in the AB, by the original Authors of Algol 60 like Peter Naur, by new

Members of IFIP's Working Group 2.1 on Algol like Tony Hoare and Niklaus Wirth, and by a large and literate group of outsiders who all felt part of the world-wide Algol movement.

Starting from the appearance of the Draft Report [MR93] as a supplement to AB 26 in February 1968, both the Authors and the Editor of the AB were deluged with mail from people who had read the document well, and gave very pertinent proposals for extensions and generalizations, as well as detailed typographical and syntactical corrections. What strikes me in retrospect is that hardly any correspondent ever suggested an important simplification of the language. Some groups of correspondents went through each iteration of the Report with a fine comb, sending in regular notes under resounding headings, like the Brussels Brainstorms, Lindsey's Lamentations, the Novosibirsk Notes, the Philips Philosophies and the Munich Meditations. Remote correspondence was followed by personal contact. Some correspondents were invited as Observers to WG2.1 meetings, and became Members of this august Committee.

In a short time, the IFIP Working Group on ALGOL developed into an intellectual conspiracy of scientists spanning (in the midst of the cold war) the West and the East, thanks to the vision of Aad van Wijngaarden and his counterpart in Novosibirsk, Andrei Petrovich Ershov.

## 2 The Authors

Professor Doctor A. van Wijngaarden, Aad for friends, the Director of the Mathematical Centre in Amsterdam, had been commissioned by WG2.1 at its Zandvoort meeting to edit the Report on a successor to Algol 60, and there was no question that he was the Boss. Short, very alert, highly dynamical, he could be the most charming person as well as the most exasperating. He introduced the formal description of the syntax by means of two-level grammars [3], which gave the Algol 68 definition much of its flavour and many of its controversial properties.

Van Wijngaarden had been joined in 1966 by Barry James Mailloux from Calgary. With his lopsided grin, crew cut and unfamiliar Canadian clothes, Barry looked like to me like a native indian. Together with his wife Isobel, he lived in an apartment right across from the Mathematical Centre. After one year in Holland, he spoke Dutch fluently<sup>1</sup>. He worked day and night, and was invaluable to van Wijngaarden.

In 1967 Van Wijngaarden was joined by John Peck from Vancouver, a quiet and dry professor with twinkling eyes. When I was drafted as an Author, John had just returned to Canada, and I first met him at the Tirrenia WG2.1 meeting, but by then I already knew his style, which found its expression in the elegant and precise syntax of the language.

Van Wijngaarden once characterized the four authors, somewhat tongue-in-cheek, as:

Koster:	transputer,
Peck:	syntaxer,
Mailloux:	implementer,
Van Wijngaarden:	party ideologist.

---

<sup>1</sup>although he never managed to hear the difference between words like 'man' and 'maan'.

Although in reality the roles were not not so clearly separated, this list does reflect the particular interests and responsibilities.

Barry Mailloux and I shared a small room on the top floor of the Mathematical Centre, overlooking the Amstel Brewery. In the mornings we worked separately or discussed together, in the afternoons we joined van Wijngaarden in the Director's office. Curiously, although all our social conversation was in Dutch, whenever we talked about Algol 68 we turned to English.

Barry was responsible for the parseability of the syntax and the implementability of the semantics of the language. At this time he was designing in detail the run-time memory management, consisting of stack and heap holding reference and procedures with different scopes. We were well aware of the existence of SIMULA 67, which gave us an existence proof, but there were at that time few implementers who had designed and implemented a system of such complexity. And Barry was interested not only in existence proofs but in efficient implementation on the hardware of the time (a modern PC is faster and has more memory than the mainframes of 1967).

He was already suffering from the illness which was to make for him the following decade into a hell of aggressive chemotherapy, and he had a terrible sense of urgency. Immediately after the acceptance of the Draft Report, the ideas he had developed about the implementation of Algol 68 were described in his Doctor's thesis [6]. A few years later, with the help of some of his students, they led to one of the few implementations of the full language Algol 68.

As the junior Author I was saddled with the definition of transput (input and output). It was an established opinion that one of the reasons for the non-success of Algol 60 outside of academic circles was the lack of well-defined transput facilities: for academics, the algorithm was all that counted, the way in which its results were transmitted to the outside world was left to the local implementation.

Both WG2.1 and the Authors were grimly decided that Algol 68 should not be scorned by Industry so easily. In order to prevent the language from going the same way as Algol 60 (see [1]), it had to surpass Fortran in its strong points: efficiency of the implementations and practicality of the transput.

When I started working, the body of the Report was there, the introductory chapters, syntax, semantics and most examples. For the transput only a rough draft existed [2], very much in the style of the "Knuthput" which had been described a few years earlier by Don Knuth for Algol 60. Barry and I realised that this approach was going to make Algol 68 neither popular nor easy to use. I started again from scratch. It was decided to first make a detailed model of the file system and devices of contemporary and imaginable computer systems, in Algol 68 itself, apart from a very few fundamentals which could not reasonably be modeled within the language.

This operational description provided also the first large-scale experience of programming in Algol 68, and gave important feedback about the pragmatic consequences of various design decisions. The work was hampered by the lack of any automated support (not even a syntax checker) and the fact that the language was in steady flux. Among other things it provided the first convincing example of a triple reference (a variable holding a mutable pointer <sup>2</sup>), a vindication of the orthogonal reference mech-

---

<sup>2</sup>See 10.5.1.2.b of [MR101]

anism.

All this was slowly taking form at the same time as the syntax of Algol 68 was being finalized and fleshed out with semantics and pragmatic remarks. The semantics was formulated in a very strict form of English, formulated so densely and polished so precisely (especially by Barry Mailloux, with Roget's Thesaurus always at hand) that I had the feeling that it would have been more merciful on the human reader to consider the sentences as formulae. In fact, the English text comes close to denotational semantics, but a suitable formalism was not yet available.

The result of all this polishing was sent out to the Computing Community as the Draft Report on the Algorithmic Language Algol 68 in February 1968.

### 3 The Fun we had

The writing of the Report was not only Work, it was also Fun, as should be apparent to all readers. It was fun to find appropriate quotations to illustrate the text, taken from authors like Lewis Carrol; the unavoidable William Shakespeare; van Wijngaarden's favourite, the Danish poet and inventor Piet Hein; A.A. Milne, who wrote some of the worlds most memorable lines about Input/Output and user interfaces; or the frivolous W.S. Gilbert.

*{Merely corroborative detail. intended to  
give artistic verisimilitude to an otherwise  
bald and unconvincing narrative.  
Mikado, W.S. Gilbert.}*

The semi-final version [MR95] even contained a quotation from the final Report, which was to follow it (R2.3.c)!

The strict and sober syntax permits itself small puns, as well as a liberal use of portmanteau words. Transput is input or output. 'Stowed' is the word for structured or rowed. Hipping is the coercion for the hop, skip and jump. MOID is MODE or void. All metanotions ending on ETY have an empty production.

*{Well, 'slithy' means 'lith and slimy'. ...  
You see it's like a portmanteau – there are  
two meanings packed up into one word.  
Through the looking-glass, Lewis Carrol.}*

Just reading aloud certain lines of the syntax, slightly raising the voice for capitalized words, conveys a feeling of heroic and pagan fun.

#### 8.6.1.1. Syntax

```
a) REFETY ROWSETY ROWWSETY NONROW slice{860a} :  
    weak REFETY ROWS ROWWSETY NONROW primary{81d}, sub symbol{31e},  
    ROWS leaving ROWSETY indexer{b,c,d,e}, bus symbol{31e}.
```

Such lines can not be read or written with a straight face.

This sense of fun carries over into the translations of the Report in other languages, like the Russian version [11], which came with a whole fresh battery of quotations.

Great ingenuity was spent on the portmanteau words, e.g. in the German version by Immo Kerner [15]. It used `MODUS` for `MODE` and translated `MOID` by `MÖSCH`:

`MÖSCH` :: `MODUS`; lösch.

leading up to `MOID` vacuum being translated as a `MÖSCH` lücke. Now it so happened that Mösch was (and is) the name of a West-Berlin construction firm, or rather a destruction firm renowned for the large holes it has made in the city of West Berlin. Not a bad joke for Eastern Germans, who were supposed to be unaware of this capitalistic hellhole!

On a more learned level, there was the problem of the unspeakable field names. In the description of the transput, structures are introduced with a number of public and secret fields. But even a secret field had to have a name. In order to prevent the programmer from accidentally discovering a secret name, in MR93 such names were made unwriteable by the artifact of starting them with an infinite number of f's. An infinite sequence of f's contains  $\aleph_0$  f's, it is all f's; so we called this sequence `ALEPH`. Since the letter Aleph was not in our type fonts, we denoted it by the percent sign `%`. In the final version of the Report, this was turned into an unwriteable letter, because the Committee didn't appreciate the joke.

## 4 The Committee

My first direct contact with IFIP Working Group 2.1, the Committee, was the meeting in Tirrenia, near Pisa, in June 1968.

At the start, the Authors were asked to summarize the reactions to MR93 that had been received, and van Wijngaarden took this opportunity to describe the changes that had been made to the language since the previous meeting (saying “The changes are based on the reactions”). The Committee was much less positive than I expected. Peter Landin deplored that the name “Algol” appeared on the document (“some of my colleagues switched from Algol to Fortran”). Gerhard Seegmüller and Brian Randell took exception to the name “Algol 68” — as if the language had already been accepted by the Committee. Van Wijngaarden was adamant and unrepentant: “I have been asked by the IFIP council to give at the IFIP Congress 68 a talk on “Algol 68””. Salt in the wounds!

Van Wijngaarden pressed for a decision by WG2.1 at the next meeting, in North Berwick. “Here is the document on Algol 68. Stamp it under the provision that after two years, in which it is planned to implement, teach and make primers, the language is subject to possible revision as result of this effort. Big companies wouldn't implement without this stamp.”

Gerhard Seegmüller then formulated four conditions for acceptance of the Report:

1. Clarification and expansion of the syntax
2. Formalization of the semantics in competitive descriptions
3. Implementation of the language
4. an Informal Introduction.

Apart from the Authors, the Committee agreed with these conditions. Van Wijngaarden, dramatically: “Does anybody know of a language of world-wide scope (such as Algol 60, Fortran, PL/1), that has been published after a compiler”?

Landin: “Lisp is an example”.

The discussion returned to technical matters. The coercion mechanism, at that time still using some explicit coercion operators, was criticized as unclear, not fundamental enough and superfluous due to the overloading of operators. The transput proposal was discussed, briefly but quite constructively, I thought. A special evening session was held with the small number of members really interested in the subject: Merner, Seegmüller, Goos and myself. The main outcome was that formats had to be made more general, including dynamic replicators.

The dynamic checking of array bounds and scopes of references was discussed. It is striking to note the great concern for micro efficiency, which has in many respects hampered the development of Algol 68. At this point the desire to avoid bound checks still led to awkward syntax, like `proc ([1:int n] real vec):`, an integer constant-declaration within a formal bound to obtain the size of the array `vec`. Similarly, a discussion of dynamic scope checks (rightly considered unavoidable in some cases), followed by a discussion on procedures delivering procedures (in some cases limited by scope problems) did not lead to the obvious conclusion to do away totally with the (statically unenforceable) scope restrictions. Algol 68, which has higher-order functions, narrowly missed having Currying, which would have made it possess a complete functional sublanguage, even though Gerhard Goos saw no problem in implementing it. In fact, the drastic and simple proposal to give every object an infinite scope was made by Hans Bekic at a later meeting (where, I do not recall), but this beautifully simple and effective proposal was not accepted for reasons of efficiency.

Another chance at more generality that was missed was the extension of operator overloading to procedures, mentioned briefly by Peter Landin. Unfortunately, the discussion veered off from this subject.

At various points in the discussion, a sore point in the description came up: the question of infinite productions (such as ALEPH), infinite modes (caused by recursive type declarations) and an infinite number of Context-Free production rules. Van Wijngaarden, who was a purely constructive mathematician, surprised me by his flippancy on the subject. When Nobuo Yoneda and Peter Landin criticized him, he responded: “This problem has puzzled us (not me). My machine may execute steps in geometrically decreasing time intervals”. Of course this would also allow “his machine” to prove or disprove Fermat’s Theorem in finite time, so this caused general laughter<sup>3</sup>.

Nobuo Yoneda deplored that the unions in MR93 were neither commutative nor absorbing, so that `union (int, real)` was not equivalent to `union (real, int)` and to `union (int, union (int, real))`. The Committee decided that unions should be made both commutative and absorbing. Van Wijngaarden protested that this was damned difficult, it would cause a terrible amount of work. Amid general catcalls that his description method was to blame, he promised a revised syntax. “We have only one life. Of course, if one of us gets ill ... – you are drawing such strong time limits on us!

---

<sup>3</sup>But I know that privately he was worried, and only articles by Lambert Meertens and myself in AB 29, showing how to deal finitely with recursive modes, put his mind at rest. In the Revised Report, this matter was resolved very elegantly.

Give me time till after lunch.” That night, he and John Peck started scribbling, and the next morning he showed us one page of syntax which solved the problem, a nice little nondeterministic automaton <sup>4</sup>.

Then came the last phase of the meeting: what would happen next. The majority of the Committee seemed to want to thank the Authors politely for their trouble, and invite others to make alternative definitions. Against this mood van Wijngaarden fought valiantly, pressing for a decision to be taken in the next meeting. At one point, he told the Committee: “This Working Group has worn out its first editor, Peter Naur. Then it has worn out two authors, Wirth and Hoare. If I understand right, it has now worn out four authors.”

Against tremendous opposition, using every rhetorical device <sup>5</sup> he managed to commit WG2.1 to a definite resolution:

The authors are invited to undertake to edit a document based on MR93, taking into account the questions and remarks received before, on, and possibly after, this meeting to the best of their power in the time they can afford for this job. This document will be submitted to the members of WG2.1 before 1 October, 68. This document will be considered by WG2.1. Either WG2.1 accepts this document, i.e. submits it to TC 2 as Report on Algol 68, or it rejects it.

Even though a large part of the meeting had been very constructive, it ended on a sour note. The behaviour of the great scientists present showed me that the progress of science is not just a matter of objective truths but also strongly influenced by human emotions. I concluded, still naively, that only a very good language defined in a very clear report could convince the members of WG2.1.

## 5 Mending the fences

We had barely 7 weeks to make a new version of the Report, for the Committee to vote on in North Berwick. In this short time we completely revised the coercion mechanism (so that all coercions were now implicit), as well as the syntax and semantics of arrays. Unions were made commutative and absorbing. Formatted transput was made much more flexible, taking up ever more pages in the Report (and code in eventual implementations).

In order to help the reader of the Report, all syntax rules were adorned with compact but helpful crossreferences. A vast number of small examples and explanations were spread as pragmatic remarks all over the text, which therefore grew appreciably in size.

All in all, I felt quite satisfied with our work, as I was driving over the appropriately named A68 to meet WG2.1 in North Berwick, with 50 copies of the revised Draft Report [MR95] in the back of my *deux cheveaux*. The Authors had done what they had been instructed to do.

---

<sup>4</sup>See 7.1.1.aa-jj and 8.2.4.a-d of [MR101]

<sup>5</sup>[7], 8.23:

SEEGMÜLLER: Then I have to vote against it!

VAN WIJNGAARDEN: I never say something final.

## 6 The North-Berwick meeting

The meeting started off badly. Since only 15 of the Workinggroup members (out of 34) had been present at Tirrenia, there had been no quorum. The validity of the Resolutions taken, and therefore even the legality of the present meeting, was in doubt. After much debate they were re-voted, and accepted. I wondered whether an illegal meeting could legalize itself by re-voting history.

Now and then, all parties took time off to blame the Chairman, Willem Louis van der Poel, for procedural errors or well-meaning remarks that managed to throw oil on the troubled flames. In fact, shouting at the Chairman seemed to be the only activity in which the WG2.1 members found themselves united.

A few hours of desultory technical discussion led to the main issue: a discussion of future work. Until then, the future had been clear: after finishing Algol X (apparently with  $X = 68$ ), the Committee would turn to the study of Algol Y, the language which was to include self-extension. Edsger Dijkstra now proposed a complicated experiment in soulsearching, which led to a heated debate: who would like to do what, and with whom? A small majority considered the finalization and maintenance of Algol 68 the most important, a large minority (including Hoare, Dijkstra and Turski) had wider plans. Actually what was happening was the birth of WG 2.3 <sup>6</sup>. The overwhelming interest was in “primitives”, i.e. elements of semantics.

Tony Hoare suggested the production of a brief document by each of the Members on each of his favourite subjects. Van Wijngaarden: “I like to thank Hoare for the distinction between the members who supply us with documents and those who do not.” This remark did nothing to clear the atmosphere.

Heinz Zemanek gave a stirring address, describing what TC2 expected from WG2.1: “You have to admit either that the document you have in your hands is the new Algol or that the editors have failed. In the second case you may charge the editors with further work or abandon the project. In the latter case you may decide that the contents is O.K. but the description has to be changed. You may also select new authors or issue the document as a preliminary one. You have, however, to make some decision, you cannot escape your responsibility, you cannot get rid of the problem.”

It was at this point that Doug Ross expressed his desire for a Minority Report, to be part of the Report. Immediately, people started discussing the modalities and timescale for preparation of such a report, rather than its desirability. Van Wijngaarden protested that at all earlier occasions, the Algol Working Committee had done without a minority report, although there was no one who agreed in every respect with the documents. But the ominous M-word was there to stay.

Discussion went on to technical subjects, array bounds and efficiency, the implementer’s burden and the Bauer Principle <sup>7</sup>. The proposal to enforce definition before application (which later led to a number of Algol 68 subsets) was rejected because it would eliminate recursive modes. The same fate befell a suggestion to consider

---

<sup>6</sup>DIJKSTRA: “The group to which I would be most attracted would be less decided by the subject of the work and more by the attitudes of other members in such a party”

<sup>7</sup>VAN WIJNGAARDEN: “Who does not want to use complex facilities, does not pay for them. If the user wants to use them, he has to pay a little.”

operators as macros rather than procedures. Micro efficiency, again. <sup>8</sup>

Back we were, on the question of decision to be taken in December, the publication of the Report and the Minority Report and the use of the name Algol 68 in courses and seminars. Most members seemed to like the language described, but this was not the case for the description. Hoare and van Wijngaarden both mentioned the possibility that the Report be published (first) under the names of the Authors, without WG2.1 responsibility, but it was preferred to accompany the Report with an eventual Minority Report, to be drawn up by the end of the next meeting.

Van Wijngaarden then brought up the fact that Peter Naur had published in AB28 a paper very critical of WG2.1 and Algol 68, sparked off by MR93. “As a Council member I will have to bring up the subject of AB28. It contains a piece of mud. IFIP pays the money for AB. I would suggest to the Council to reconsider the money appropriated for editing AB.” He was all the more offended, because it was reproduced and distributed by the Mathematical Centre. Why he chose to put this matter before WG2.1, I do not know. His remarks were of course very unfair to Frazer Duncan, the Editor of the AB, and led to a heated discussion, covering most of the morning, regarding censorship, refereeing of articles and duties towards the Computing Community.

Van Wijngaarden could make himself either greatly liked or immensely unpopular. Barry Mailloux had to say “Chucks, fella’s” many times in his most reconciliatory tone before order was restored. Later, in the absence of van Wijngaarden, Barry felt it necessary to apologize for him. “He did not refuse to reproduce AB, he wanted to express his personal dislike, but nothing more. The second thing is that his statement “I am offended” is a technical and not a personal remark. On my own behalf, I would like to say the following: it was suggested that I and some other people seek fame, credit and fortune. I would like to deny it. I did a lot of work because I do believe in the language and believe strongly enough to wish to propagate it.”

Thus ended the meeting, which had brought little technical progress but which had prepared the stage for the drama to be enacted in Munich, in December 1968.

## 7 The IFIP Congress

The IFIP 1968 Congress took place that August in Edinburgh, just a few hours drive away from North Berwick. Van Wijngaarden’s invited lecture on Algol 68 was to me the high point of the conference, and not only to me. The auditorium was packed, people were standing on all sides, even in the corridors and outside, in front of the hall. Van Wijngaarden appeared in the centre, smiling radiantly. “Let me sell you a language”, he started, and proceeded to outline the ideas behind the language. He showed some examples. “Can you define triangular arrays?” someone (Tony Hoare?) interrupted. “Not just triangular, but even elliptical” replied Aad, and showed how. He carried the listeners with him, from scepticism to enthusiasm. There was a prolonged applause.

---

<sup>8</sup>VAN WIJNGAARDEN: “... If we accept this point of Mr Lindsey we will produce a FORTRAN-like language, by which I mean its intellectual level.”

RANDELL: “It is well to remember that there are also good sides in the FORTRAN intellectual level – do not forget its efficiency.”

Vehemently discussing, people streamed out of the hall. A small man pushed through the throng, straight at me. “Conkratulations, your Master hass done it” said Niklaus Wirth in his inimitable Swiss-German English.

## 8 Towards Munich

At Munich, it was then, that the ultimate choice would be made. Again Van Wijngaarden, Mailloux and I went over the whole text, making the last changes in the language and its description, cleaning up various dark formulations, correcting small errors and in the process retyping everything. Van Wijngaarden loved the freedom in typefonts offered by the new IBM “golfball” printer, and introduced outlandish symbols for various operators with boldface names (e.g.  $\square$  for **ELEM** and  $\lceil$  for **UPB**). He found a use for every symbol on the APL-typeball. He would have loved  $\text{\TeX}$  and the possibilities to define new typefonts!

There existed no wordprocessing software to speak of, and we had not even the support of an editor to mechanize the production of the Report. How many times have I glued small strips of white paper over Snopake-encrusted originals? By now even sometimes the wording of sentences was influenced by the fact that they had to fit within a given space. The nearer the deadline, the more frantic the work became. We were joined by Lambert Meertens, but still things went too slowly. The text kept changing, always for good reasons, and there was no chance to leave the normal period for an orderly offset production. The printer taught us to make matrices for the offset machine. In the end we had to learn how to bind and glue the whole document. Van Wijngaarden took time off to design and produce a suitable cover. It showed a pattern built out of hundreds of elem-signs — and one little commercial at-sign, his personal mark.

On the morning of the last day, after a frantic night and just before our flight left, the work was finished: we had produced the first printing of the Final Draft[10]. No time to catch up on sleep. Lambert and I found ourselves sitting in front of the plane, dog tired. What would the Committee decide? We were too tired even to speculate. The stewardess brought us, unbidden, two baby bottles of champagne each, from a Gentleman in the back. We looked over the back of our seats: Van Wijngaarden sat there, besides a sleeping Barry Mailloux, prim as a daisy, and waved his hand at us. The Authors had done their job.

## 9 And after

Algol 68 was accepted by WG2.1 as its own child at the Munich meeting in December 1968, but it was a Pyrrhus victory for van Wijngaarden: a large minority dissented, and wrote a minority report.

Translations of the Report in many languages appeared [11, 13, 14, 15], as well as an Informal Introduction [12] and textbooks explaining the two-level formalism [16]. Implementations were slow in coming, apart from a number of official subset implementations [A68R, A68S], limited but successful. Before implementations of the full language became available, the state-of-the-art in compiler making had to be advanced

quite a lot. The language was used in many courses. Its effect, through teaching, on the minds of a generation of computer scientists was much greater than its utility in practical applications.

The announced Revision of Algol 68 started almost immediately and took until 1974 [RR]. It resulted in an exemplarily clear, precise and consistent description of an elegant and orthogonal language that was at its appearance already classical but dead — the Ilias and Odyssee of Computer Science.

What has gone wrong? Many convincing reasons can be found. The lack of timely implementations (but those people who have actually programmed in Algol 68 remember it as a beautiful means of expression); the obscurity of the description (which is denied by virtually anyone who has bothered to study it); the lack of political and industrial backing (the fate of Algol 60, all over again). I think that Algol 68 was the result and the conclusion of a decade of search for the ideal Algorithmic Language, but that the time for universal programming languages was already over when it appeared. In the seventies, research went on to other problems: Software Engineering, System Implementation Languages, Databases and Computer Graphics.

The two-level grammars, used in the syntactic description of Algol 68 have, through the Metamorphosis Grammars of Colmerauer, led to PROLOG and Logic Programming. The spirit of orthogonality and elegance that characterizes Algol 68 influenced the design of many functional languages. Algol 68 lives on, not only in the minds of people formed by it but also in very unlikely places, like C and C++, whose concepts and terminology at numerous places give a weird echo of Algol 68. Even though there the orthogonality and elegance of the syntax and the security of the semantics have mostly been lost, a whole new generation of programmers uses coercions and casts. In fact, the boisterous discussions in the present programming community about the shortcomings of C++ and solutions to overcome them gives me a strong feeling of déjà vu, reminding me of the making of Algol 68. The Far West of Computer Science.

In 1974, during an IFIP WG2.4 Meeting in Berlin, I was stopped in the corridor by Jean Ichbiah, the author of the language LIS and designer of what was to become ADA, on his way to the success that brought him the Legion d'Honneur. He said to me with great emphasis: “We are going to do right what Algol 68 has done wrong”. Have they really, I wonder?

## References

- [1] R.W. Bemer, *A politico-social history of ALGOL*. In: *Annual Review in Automatic Programming* 5, 1969.
- [2] J.V. Garwick, J.M. Merner, P.Z. Ingerman and M. Paul, *Report of the Algol - X - I - O Subcommittee*, WG2.1 Working Paper, July 1966.
- [3] A. van Wijngaarden, *Orthogonal Design and Description of a Formal Language*, Mathematisch Centrum, MR 76, October 1965.
- [4] A. van Wijngaarden and B.J. Mailloux, *A Draft Proposal for the Algorithmic Language Algol X*, WG 2.1 Working Paper, October 1966.
- [5] A. van Wijngaarden, B.J. Mailloux and J.E.L. Peck, *A Draft Proposal for the Algorithmic Language Algol 67*, Mathematisch Centrum, MR 92, November 1967.

- [MR93] A. van Wijngaarden (Editor), B.J. Mailloux, J.E.L. Peck and C.H.A. Koster, *Draft Report on the Algorithmic Language Algol 68*, Mathematisch Centrum, MR 93, January 1968.
- [6] B.J. Mailloux, *On the implementation of Algol 68*, Ph.D. thesis, University of Amsterdam, June 1968.
- [7] Informal Minutes of the IFIP WG2.1 Meeting in Tirrenia, June 3 - 7, 1968.
- [MR95] A. van Wijngaarden, B.J. Mailloux, J.E.L. Peck and C.H.A. Koster, *Working Document on the Algorithmic Language Algol 68*, Mathematisch Centrum, MR 95, July 1968.
- [8] Informal Minutes of the IFIP WG2.1 Meeting in North Berwick, July 28 - August 1, 1968.
- [9] A. van Wijngaarden, B.J. Mailloux, J.E.L. Peck and C.H.A. Koster, *Penultimate Draft Report on the Algorithmic Language Algol 68*, Mathematisch Centrum, MR 99, October 1968.
- [10] A. van Wijngaarden (Editor), B.J. Mailloux, J.E.L. Peck and C.H.A. Koster, *Final Draft Report on the Algorithmic Language Algol 68*, Mathematisch Centrum, MR 100, December 1968.
- [MR101] A. van Wijngaarden (Editor), B.J. Mailloux, J.E.L. Peck and C.H.A. Koster, *Report on the Algorithmic Language Algol 68*, Mathematisch Centrum, MR 101, February 1969; also *Numerische Mathematik*, Vol 14, pp 79-218, 1969.
- [11] A.A. Bährs, A.P. Ershov, L.L. Zmieskaya and A.F. Rar, *Soobshchenije ob algorimicheskom jazyke Algol 68*, *Kybernetika*, Kiev, Vol. 6, 1969 and Vol. 1, 1970.
- [12] C.H. Lindsey and S.G. vander Meulen, *Informal introduction to Algol 68*, North Holland Publishing Company, Amsterdam 1971.
- [13] D. Toshkov and St. Buchvarov, *Algoritmichniyat yezik Algol 68*, *Nauka i Yzkustvo*, Sofia 1971.
- [14] J. Buffet, P. Arnal and A. Quere, *Definition du Language Algorithmique Algol 68*, Hermann, Paris, 1972.
- [15] I.O. Kerner, *Bericht über die Algorithmische Sprache Algol 68*, Akademie-Verlag, Berlin 1972.
- [RR] A. van Wijngaarden, B.J. Mailloux, J.E.L. Peck, C.H.A. Koster, M. Sintzoff, C.H. Lindsey, L.G.L.T. Meertens and R.G. Fisker, *Revised Report on the Algorithmic Language Algol 68*, *Acta Informatica*, Vol. 5 Tome 1-3, Springer-Verlag, Berlin, 1975.
- [16] J. C. Cleaveland and R.C. Uzgalis, *Grammars for Programming Languages*. Elsevier/North Holland, 1977.
- [A68R] I. Currie et al, Algol 68R, Royal Radar Establishment, Malvern UK.  
Presently available from  
<ftp://ftp.dra.hmg.gb/pub/ella>
- [A68S] P. Hibbard et al, Algol 68S, Liverpool University, Carnegie Mellon University  
An implementation by C.H. Lindsey is resently available from  
<ftp://ftp.cs.man.ac.uk/pub/ch1/A68S>