



UvA-DARE (Digital Academic Repository)

From Universal to Programming Languages

Gobbo, F.; Durnová, H.

[Link to publication](#)

Citation for published version (APA):

Gobbo, F., & Durnová, H. (2014). From Universal to Programming Languages. Amsterdam: University of Amsterdam, Amsterdam Center for Language and Communication [etc.].

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <http://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

From Universal to Programming Languages

Federico Gobbo¹ and Helena Durnová²

¹ University of Amsterdam, Amsterdam Center for Language and Communication,
Spuistraat 210 NL-1012 VT Amsterdam, The Netherlands

`F.Gobbo@uva.nl`

² Masaryk University, Faculty of Education, Department of Mathematics,
Poříčí 31 CZ-603 00 Brno, Czech Republic

`helena.durnova@mail.muni.cz`

Abstract. In the history of computation, the reflection over language plays an important role in its foundational days, still to be fully investigated. In particular, the effort to find a perfect, universal language apt to sustain transnational communication among scientists was often directed towards the reduction of semantic ambiguity and cultural neutrality. The result is a class of non-natural languages created in the same period and sometimes by the same scientists involved in logic, mathematics, and computability, such as Descartes, Leibniz and Peano. Finally, as a special case, we will analyse the use of the metaphor of Esperanto within the history of Computer Science.

Keywords: universal languages, programming languages, Esperanto

1 Introduction

History of computation and history of language and communication are intertwined from the early days, which is also reflected in the common etymological origins of the the words *reckon*, *count*, and *compute*. These interconnections are apparent not only in English, but also in other languages: in Spanish *contar* means both narrate and compute, in German *zählen*, to calculate, and *erzählen*, to narrate, have common roots, and in modern French, *compter*, to count, and *conter*, to tell, are distinct, but have common etymological origin. Moreover, the word ‘language’ is used abundantly in contemporary Computer Science and related disciplines. Some examples are: SQL, for Structured Query Language; XML, for eXtensible Markup Language; PSL, for IEEE 1850 Standard for Property Specification Language. Of course, it appears frequently in the common phrase ‘programming language’. All these languages are different in nature from languages used for communication between humans, being tailored for human-machine and machine-machine information exchange. Nonetheless, the widespread use of the word ‘language’ among computer scientists and informaticians has invited analogies with terms used in linguistics, such as ‘syntax’ and ‘semantics’ of programming languages. Moreover, metaphors were borrowed from natural languages: e.g., the steep rise in the number and variety of programming languages was described by a Tower of Babel – see Hellige [18].

Martin Davis [8] describes the origins of modern computation as a road from Leibniz to Turing. Intriguingly, the main figures of this history started their reflection on the fact that natural languages are ambiguous in semantics and redundant in syntax, and they tried to find an optimal way to maximize expressivity reducing ambiguity and redundancy to a minimum via formal, non-natural language systems. For instance, George Boole introduced his laws of human thought through an example in English: ‘if x alone stands for “white things,” and y for sheep, let xy stand for “white sheep;”’ – in Davis [8, 28].

In this paper, we will show how the quest of universality in languages proceeded in parallel with the foundation of computation. In particular, the final part of this paper will explore the case study of Esperanto, the most widely known attempt to establish a universal language, its metaphorical use by computer specialists, and the similarities of its history with one of the foundational efforts in computer science, namely the late 1950s effort to create a universal programming language. We will suggest that in the late 1950s, computer programmers were looking for their own Esperanto with ideals and enthusiasm similar to the earlier attempts to find a universal language for human communication. Since many people doubt the success of Esperanto, the metaphor may be interpreted in both positive and negative sense; in any case, this quest for a unifying programming language bears some similarities to the earlier attempts to define a universal language.

2 Universal languages and the scientific revolution

The discovery of the ancient Egyptian and Chinese civilizations greatly influenced English philosophers of the 17th century. Bacon studied hieroglyphs and Chinese writing, formulating the concept of *real characters*, “artificially constructed characters, whose significance depended only on a custom or usage which was arbitrarily established (*ad placitum*) and agreed upon ‘as though by a silent pact’ (*tanquam pacto tacito recepti*)” – see Rossi [37, 146]. When real characters are put together, they can form an artificial language free from semantic ambiguities, as characters should represent reality directly as concepts instead of sounds. This class of artificial, philosophical languages were called ‘perfect’ or ‘universal’ by their proponents, which are two key features shared by all the members of the class. While ‘perfection’ is an internal feature, i.e., their crystal clear shape, often described in logical terms, ‘universality’ is an external feature, being neutral in ethnic and cultural terms. Some of the most important attempts to find the definitive philosophical language after Bacon’s idea were proposed by Francis Lodowick, George Dalgarno, and most notably John Wilkins – see at least Eco [10] and Maat [28]. Another perspective was taken by Comenius (1592–1670) who in his philosophical work *Via Lucis* dreamt for a language that would heal the world – see Durnová [9] and Formizzi [11]. Universal language was a road to peace, and although Comenius did not define this language, he determined what qualities such a language should have. Noticeably, he also suggested keeping all the existing languages used by learned and/or religious

people (Hebrew, Greek, and Latin) and called for the cultivation of vernacular languages as well. In Comenius’s time, the notion of universal language overlapped with the notions of protolanguage and perfect language – see Pavlas [33]. Descartes also discussed the realizability of the universal language in a letter to Mersenne on 20 November 1629, in which the French philosopher accepted the idea of decomposing concepts over reality in ‘simple ideas’ which should be combined afterwards, in a way analogous to arithmetics. Descartes’s atomism was rejected by Leibniz, who developed the infinitesimal calculus and believed that there is no semantic primitive in the universe – see Marrone [29]. Rather, a rational calculus (*calculus ratiocinator*) should be found to avoid ambiguities in philosophical disputes, as is expressed in the motto *calculemus!*, let’s calculate!. Leibniz called his philosophical language ‘universal characteristic’ (*characteristica universalis*), in direct contrast with Wilkins’ real characters. In other words, symbols (‘characters’) should represent the alphabet and calculational tools of human thinking, which is ‘universal’ instead of ‘real’. Davis [8] claims that Leibniz’s idea influenced logicians such as George Boole and Gottlob Frege, eventually giving the foundations of Hilbert’s program. However, even if Leibniz’s ‘general language’ (*lingua generalis*) was composed in 1678, his work deeply influenced European mathematicians and logicians only after the publication of previously unknown works, edited by Louis Couturat [6] in the early days of the 20th century.

3 Logicians and Universal Languages in the 20th century

In the year 1900, Paris hosted both the conference where Hilbert presented the research programme for the mathematics in the new century and the international congress of philosophy. There were three people who attended both events: Bertrand Russell, Louis Couturat and Giuseppe Peano. It was on that occasion that Russell (by far the youngest of the three) got acquainted with the mathematical logic of Peano and his school, while Couturat presented the movement for the adoption of an international auxiliary language to the community of philosophers as the descendant of the ideas of Leibniz and the English philosophers of the 17th century – see Gray [16], Chapter 6. In the years 1900-1907 Couturat was active in the Esperanto movement, and in that period he gathered the support of 310 academic and professional societies, and 1,250 university professors and scholars – Couturat in Couturat *et al.* [5]. He lead this support through a Delegation, which decided that Esperanto should be reformed in order to be adopted as the definitive form of the international auxiliary language – see Large [23] for details.

Couturat exchanged a lot of letters with important colleagues of his time: Gottlob Frege, Henri Poincaré, Émil Borel, Fortunately, Couturat’s correspondence with Russell and Peano survived almost complete on both sides, and thus we can see that two main topics were common in both exchanges of letters: first, logic and the foundations of mathematics, second, international auxiliary languages – see respectively Schmid and Carlevaro [38] and Luciano and Roero

[27]). While both correspondences started more or less at the same time (with Peano in 1896, with Russell in 1897) Peano abruptly stopped writing to the French philosopher in 1910, while Russell’s correspondence with both Couturat and Peano dwindled in 1908. In those years, Russell was probably heavily involved in working on the monumental work *Principia Mathematica* with A. N. Whitehead. In 1908, Peano published the last edition of his *Formulario Mathematico*, written in *Latino sine Flexione*, a simplified version of Latin heavily influenced by Leibniz’s *lingua generalis*, and in the same year, Couturat formed a movement in support of Ido, a reform of Esperanto, claiming it to be the result of the work by the Delegation, supported by eminent professors at that time – see Lorenz in Couturat *et al.* [5]. While Peano and Couturat were active supporters of this struggle, Russell was sceptical. After the sudden death of Couturat in 1914 and the Great War, the interest by logicians and mathematicians in the proposals against Esperanto gradually faded out; similarly, Peano’s death in 1932 caused the decline of his simplified version of Latin. In the same years, Gödel’s results led research to the fundamental results in mathematics and mathematical logic by Alan Turing, Alonzo Church and Emil Post in 1936. The struggle to find the definitive Universal Language was a failure that had interesting side effects, while the failure of Hilbert’s program eventually led to the Universal Turing Machine.

4 Taxonomies of universal languages

It is important to note that there is a change of focus from the 17th to the 20th century among logicians and mathematicians: if the pioneers of the scientific revolutions granted a privilege to the modelling of human reasoning, logicians of the past century had a pragmatic focus, i.e. solving international communication through a language to be used as a natural one, both in speech and in writing. For this reason, the need of a classification of non-natural languages emerged in the early days of the 20th century. In 1903, Couturat and Leau [7] offer a survey of almost all universal languages proposed previously and propose a distinction that achieved high credit among scholars of that time. Artificial languages can be divided into three categories: a-priori, mixed and a-posteriori. A-priori languages are the artificial languages built over one or more abstract principles of construction: this class includes all pasigraphies, i.e., writing system where each symbol represents a concept, following the Baconian idea of “real characters” already illustrated in Section 3. Okrent [32] lists approximately 500 pasigraphies. The most relevant mixed language is Volapük, the first international auxiliary language to be used in practice – see Hauptenthal [17]. Although it is partially based on German, English, and other languages, in Volapük word compounding is partially a-priori, which made the language became difficult to master. For example, the compound *nobastonacan* for ‘jeweller is formed by ‘nobility’ (*nob*), ‘stona’ (*stone*) and ‘merchandise’ (*can*), with the genitive ending *a* in-between – see Couturat and Leau [7], p. 141. On the contrary, when an international auxiliary language clearly borrows its structure from an existing natural language,

the result is an a-posteriori language: Zamenhof's Esperanto and Couturat's Ido are important members of this class.

While fortunate, the taxonomy by Couturat and Leau has severe limits: in particular, it does not take into account clearly artificial languages as for instance first-order logic, nor does it explain the difference from clearly natural languages such as English, Urdu or Chinese. Even if alternative taxonomies of universal and international languages do exist – e.g., Blanke [4] and Gobbo [15] – they fail to give a comprehensive framework to the phenomenon of non-natural languages in general. The only comprehensive taxonomy in the literature, at least to the extent of authors' knowledge, is the one proposed by Lyons [26], which articulates the dichotomy 'natural *vs.* artificial languages' in four levels of (non) naturalness. Natural₁ languages are conform with nature: French, Arabic are implicitly classified as natural₁ languages being learnable (and learnt) as mother tongues. Natural₂ languages are constrained by nature, i.e., they are species-specific: in Chomskyan terms, all human languages are natural₂ being instances of the Universal Grammar. It follows that natural₁ languages are included in natural₂ languages. Natural₃ languages are acquired by humans as a normal part of the process of maturation and socialization: sign languages such as the American Sign Language (ASL) and the British Sign Language (BSL) are natural₃. The weakest level is naturalness₄, which depends on the specific judgement of naturalness by a concrete human being: for example, Montague [39] describes his theories and descriptions of language as 'natural'. In naturalness₄, the property refers to the theory of language built over the language instead of the language itself.

Following his taxonomy of naturalness_{1,2,3,4}, the author defines two different kinds of artificial languages: unnatural₃ languages (implying unnaturalness_{1,2}) and unnatural₄ languages. Unnatural₃ languages comprise the ones designed by logicians, mathematicians and computer scientists: post-Baconian a-priori universal languages, Boolean and predicate calculi, as well as all Turing-complete programming languages. On the other hand, unnatural₄ languages comprise Couturat and Leau's a-posteriori languages such as Esperanto as well as Quasi-Natural Languages (QNL) commonly constructed by linguists by deliberately changing their structural properties for experimental purposes – for a discussion, Gobbo [14].

5 The Case-Study of Esperanto in Computer Science

As we have seen in the previous section, at the turn of the 20th century the idea of an international auxiliary language, and the Esperantist movement in particular, attracted part of the European logicians and mathematicians – see Chapter 6 in Gray [16]. Computer scientists looking for their own roots tend to say that the theoretical foundations of Computer Science were posed in the same period, alongwith their first reflections on Artificial Intelligence. However, in his 1972 paper, Donald Knuth [22] traced the history of computer science back to Babylonian mathematics and called ancient Babylonian mathematicians the first

programmers. Sometimes these two lines of research – Esperanto and Computer Science – were present in the same persons, even if apparently unrelated. In particular, in 1915 the Spanish scientist Leonardo Torres y Quevedo – an active Esperantist, see Barrio [1] – invoked a new science called *automatique* (French word for ‘automatics’), where an idea of artificial intelligence was proposed [12]. Quevedo’s work influenced Norbert Wiener, the founder of Cybernetics, whose father, Levi Wiener, had been a Warsaw Gymnasium schoolmate of Zamenhof, the founder of Esperanto, and himself an active Esperantist in the 1930s – see Wiener [40], p. 13.

Before Computer Science started to develop into a mature discipline in the second half of the 20th century, no proposal of international auxiliary languages succeeded in forming a stable speech community surviving two World Wars, save Esperanto. Esperanto, it should be noted, survived despite having been explicitly persecuted by totalitarian regimes ruled by Hitler and Stalin – see Lins [25]. It is thus not surprising that after the Second World War the word ‘Esperanto’ became a metaphor for ‘universal language’ and ‘auxiliary language’. In Computer Science, Esperanto has been used both as a metaphor and as a tool. In particular, we want to highlight the use of Esperanto in the fields of machine translation and in the early days of programming languages.

5.1 Machine Translation and Esperanto

Translation is the act of rendering a text written in a natural language into a reliable text written in another natural language, where the meaning of the original text should be preserved. Machine translation means that the translation is carried out by a computer without any human aid. However, in principle the translating machine can be any mechanical device, not necessarily a digital computer. There are two experiences where Esperanto played a significant role as the main tool for machine translation, the first one with a mechanical device, the second one with a digital computer.

The first experience goes back to the 1930s. One of the pioneers of mechanical translation is the Soviet Union scientist Petr Petrovich Smirnov-Troyanskii – see Hutchins and Lovtskii [19]. In 1933 he obtained a Soviet patent for a mechanical machine for translation, made of a special desk with a typewriter, a photographic camera and a belt. Initially, he borrowed the symbols of parsing from Esperanto, while afterwards Troyanskii changed them because Esperanto became *lingua non grata* in the USSR. However, his work did not have any effect on the history of machine translation, which dates from a memorandum written by Warren Weaver in 1949 in the United States – for a review, see Hutchins [21].

The second experience was a specific project, called Distributed Language Translation (DLT), where Esperanto played a key role – see the memorandum by Witkam [41]. The original idea was presented by Witkam in 1980 in the International Conference on New Systems and Services in Telecommunications, in Liège, Belgium. In the early 1980s, Japan had launched the Fifth Generation Computers project, and the European Commission was ready to fund innovative projects in the field of machine translation. DLT was successfully funded for a

feasibility study in the years 1982-3. The engine was programmed in Prolog, a logic programming language based on unification very popular in those years. After many years of development, a prototype was presented in 1987. While English and French were the natural languages involved, the translation engine was based on Esperanto. Since 1988, Esperanto was left out in order to prepare the commercial version, eventually on the market in 1993. However, undoubtedly Esperanto was central in DLT – see Chapter 17 in Hutchins and Somers [20].

5.2 ALGOL 60 and the need for a ‘Programmer’s Esperanto’

The situation of computer programmers in the mid-1950s was somewhat similar to the situation of scientists at international congresses at the turn of the 20th century. In the following, we will try to explain in what ways the ALGOL effort is comparable to the Esperanto movement in the early 20th century. We will concentrate on the sociological, rather than technical, aspects of the development of ALGOL 60, and while our focus is different, our story line may appear different from the one presented by the participants in the ALGOL effort, most notably (in alphabetical order) Friedrich Bauer, Peter Naur, and Alan Perlis. We believe that focusing on the intellectual devotion invested in the design of ALGOL bears some comparison with the earlier searches for universal languages.

Before showing the similarities between Esperanto and ALGOL, we also need to point out that we will not be discussing the role of ALGOL 60 on the development of compilers. These issues have recently been covered e.g. by Nofre *et al* [30]. Rather, we would like to suggest that the dynamics of the group of computer scientists working on ALGOL 60 resembles the Esperanto movement.

One of the issues that became visible in mid-1950s was that programming, an activity that had once been thought about as relatively easy (especially the part of coding the numerical procedure for the computer), was also relatively boring and prone to making mistakes. So, several attempts were made to make this process more controllable through automation through systematic description of the numerical procedure. In 1955, at the conference in Darmstadt, Heinz Rutishauser, a Swiss mathematician and a pioneer in numerical analysis, and his (East-)German colleague N. Joachim Lehmann both called for universal notation for the description of the computing process – see Joachim Lehmann [24], pp. 256–258. Alwin Walther, an organizer of the Darmstadt conference, supported this thought and pleaded for such a way of writing computer programs that would allow for a program designed in Zürich to be immediately legible in Darmstadt. In Darmstadt, computing specialists from various countries agreed that they needed to communicate more efficiently. Their situation and the solution stemmed from the need to share computer programs between different computers and computing centres – see Nofre [31]. Eventually, in 1957, the American ACM (Association for the Computing Machinery) and the German GAMM (*Gesellschaft für angewandte Mathematik und Mechanik*) prepared proposals for a Universal Language, called *Proposal for a Programming Language* and *Proposal for a universal language for the description of computing processes*, respectively. The adjective ‘universal’ was adopted then easily by Poyen and Vauquois [36].

Eventually, this led to a situation similar to the search of a universal language: there were some reservations in choosing one of the existing languages as a standard, so the ensuing effort to design a new programming language bears resemblance to much earlier efforts to design a language for human communication at the turn of the 20th century. In the past, such languages were called universal or auxiliary and they were supposed to make the world a better place to live: similarly, looking at the first proposals for an international standard for algebraic language in programming (Perlis and Samelson [34]), for a universal language (Bauer *et al.* [2]), or for an international algebraic language (Perlis and Samelson [35]), one can see the attempts to make programming easier through introducing a language that would meet all the qualities. The effort seems to have been accepted as such and it is visible in the overall support for using ALGOL in the scientific computing circles. Furthermore, in a lecture to undergraduates in 1960, Antonín Svoboda – the Czechoslovak computer pioneer – named ALGOL the “Programmer’s Esperanto”, a language every programmer should learn. Accordingly, the language was taught to all Czechoslovak students at university level. In fact, in 1961 Ginsburg and Rice [13] proved that ALGOL is not an unicum but rather a member of a *family* of formal, artificial languages. From then on, it would be possible to produce a new Babel of programming languages, where their computational power – equivalent to a universal Turing machine – will be guaranteed by their description in terms of context-free grammar expressed via the Backus-Naur normal form. On the other hand, linguists interested in the formalization of natural languages will turn their attention to context-sensitive grammars, another level in the Chomskyan hierarchy, as context-free grammars quickly proved to be insufficient to represent natural languages in a formal way.

6 Conclusion

The struggle for a perfect, universal language – both in the case of humans and machines – is doomed to fail. Today, few people know about Ido and the other concurrent auxiliary language projects alongside Esperanto, and the *de facto* international language is English. On the other hand, instead of having one definitive programming languages computer programmers should choose the one to be used among hundreds if not thousands of them, each having its own specific advantages and disadvantages. However, as claimed by Eco [10], p. 19: ‘the story of the search of the perfect language is the story of a dream and of a series of failures. Yet that is not to say that the story of failures must itself be a failure.’ Without the dream of the universal language, perhaps side-effects would not be found. In the 20th century, the metaphor of universality is not the only one used by scholars involved in the foundation of computability and in the auxiliary language movement. In fact, the idea of internationality and of planning should be added. Notably, Couturat *et al.* [5] refer to internationality in the title of their work, while the published sources on the ALGOL effort point to establishing an international standard. The parallels between *Plansprachen*, lit. ‘planned

languages’ – term coined by Eugen Wüster, pioneer scientific terminology standardization but also of Esperantic studies, see Blanke [3] – and Konrad Zuse’s *Plankalkül*, lit. ‘planned calculus’, could also be explored in further research.

Acknowledgements

Federico Gobbo is appointed at the University of Amsterdam on behalf of the Universal Esperanto Association (UEA, Rotterdam). Usual disclaimer applies.

References

1. del Barrio Unquera, J. A.: Leonardo Torres Quevedo y el esperanto. In González de Posada, F.: Leonardo Torres Quevedo: conmemoración del sesquicentenario de su nacimiento (1852). Sociedad Estatal de Conmemoraciones Culturales. (2003)
2. Bauer, F. L.; Bottenbruch, H.; Rutishauser, H.; Samelson, K.: Proposal for a universal language for the description of computing processes. In: J. W. Carr (ed.), Computer Programming and Artificial Intelligence, University of Michigan Summer School, 355–373. (1958)
3. Blanke, W.: Esperanto - Terminologie und Terminologearbeit. Mondial. (2008).
4. Blanke B.: Internationale Plansprachen. Eine Einführung. Akademie Verlag (1985).
5. Couturat, L.; Jespersen, O.; Lorenz, R.; Ostwald, W.; Pfaundler, L.: International Language and Science: Considerations on the Introduction of an International Language into Science. Constable & C. Ltd. (1910).
6. Couturat, L.: Opuscles et fragments inédits de Leibniz. Félix Arcan Éditeur. (1903).
7. Couturat, L.; Leau, L.: Histoire de la Langue Universelle. Hachette. (1903)
8. Davis, M.: The Universal Computer: The Road from Leibnitz to Turing. W.W. Norton. (2000)
9. Durnová, H.: Programming language ALGOL 60 as a new Esperanto? XXVIII International Colloquium, Brno, May 20 (2010)
10. Eco, U.: The Search for the Perfect Language. Wiley-Blackwell. (1997)
11. Formizzi G.: Comenio. InKoj. Philosophy & Artificial Languages 3(2), 167–180. (2012)
12. Garrido M.: Torres Quevedo y Alan Turing: dos visiones de la tecnología de la mente. In González de Posada, F.: Leonardo Torres Quevedo: conmemoración del sesquicentenario de su nacimiento (1852). Sociedad Estatal de Conmemoraciones Culturales. (2003)
13. Ginsburg, S.; Rice, H.G.: Two Families of Languages Related to ALGOL. Journal of the ACM 9(3), 350–371. (1962)
14. Gobbo F.: Alan Turing Creator of Artificial Languages. InKoj. Philosophy & Artificial Languages 3(2), 181–194. (2012)
15. Gobbo, F.: Planning languages and language planning: e contribution of Interlinguistics to cross-cultural communication. In Boers, F.; Darquennes, J.; Kerremans, K.; Temmerman, R. (eds): Multilingualism Applied Comparative Linguistics. Volume II. Cambridge Scholars Publishing. (2008)
16. Gray, J.: Plato’s Ghost: The Modernist Transformation of Mathematics. Princeton University Press. (2009)
17. Hauptenthal, R.: Volapük die Weltsprache: Entwurf einer Universalsprache für alle Gebildete der ganzen Erde. Hildesheim. (1982)

18. Hellige, H.-D.: The metaphor of Babel in computing. Paper given at *Maschinen mit Phantasie: Festsymposium zum 90. Geburtstag von Heinz Zemanek*, Mnchen (22.2.2010).
19. Hutchins, J.; Lovtskii, E.: Petr Petrovich Troyanskii (189–1950): A Forgotten Pioneer of Mechanical Translation. *Machine Translation* 15, 187–221. (2000)
20. Hutchins, J.N.; Somers, H.L.: *An Introduction to Machine Translation*. Academic Press. (1992)
21. Hutchins, W.J.: *Machine Translation and Machine-Aided Translation*. *Journal of Documentation*, 34(2), 119–159. (1978)
22. Knuth, D.E.: Ancient Babylonian algorithms. *Communications of the ACM*, 15(7), 671–677. (1972)
23. Large, A.: *The Artificial Language Movement*. Basil Blackwell. (1985)
24. Lehmann Joachim, N.: ALGOL im Ostblock und der Weg zu Systemen von Programmiersprachen. In Hellige, H.-D. (ed.), *Geschichten der Informatik: Visionen, Paradigmen, Leitmotive*. Springer. 255–274 (2004)
25. Lins, U.: *Die gefährliche Sprache: Die Verfolgung der Esperantisten unter Hitler und Stalin*. Bleicher. (1988)
26. Lyons, J.: *Natural Language and Universal Grammar*. Vol. 1. Cambridge University Press. (1991).
27. Luciano, E.; Roero C. S.: *Giuseppe Peano–Louis Couturat (1896–1914)* Olschki. (2005)
28. Maat, J.: *Philosophical Languages in the Seventeenth Century: Dalgarno, Wilkins, Leibniz*. Kluwer. (2004)
29. Marrone, C.: *Le lingue utopiche*. Stampa alternativa. (2004)
30. Nofre, D.; Alberts, G.; Priestly, M.: When Technology Became Language: The Origins of the Linguistic Conception of Computer Programming, 1950-1960. *Technology and Culture*, 55(1), 40–75.
31. Nofre, D.: Unravelling Algol: US, Europe, and the Creation of a Programming Language. *IEEE Annals for the History of Computing*, 32(2), 58–68. (2010)
32. Okrent, A.: In *The Land of Invented Languages*. Spiegel & Grau (2009)
33. Pavlas, P.: *Idea dokonalého jazyka u F. M. van Helmholtza a J. A. Komenského*. *Studia Comeniana et Historica* 43:89-90, 286–305. (2013)
34. Perlis, A.J.; Samelson, K.: Report on a Proposed International Standard for a Common Algebraic Language for Digital Computers. In: J. W. Carr (ed.), *Computer Programming and Artificial Intelligence*, University of Michigan Summer School, 375–401. (1958)
35. Perlis, A.J.; Samelson, K.: Preliminary report: international algebraic language. *Communications of the ACM*, 1(12), 8–22. (1958)
36. Poyen, J.; Vauquois, B. A propos d’un langage universel. *International Conference on Information Processing, UNESCO, Paris*. (1959)
37. Rossi, P.: *Logic and the Art of Memory: The Quest for a Universal Language*. Continuum. (2000)
38. Schmid, A.-F.; Carlevaro, T. (eds.): *Bertrand Russell. Correspondance sur la philosophie, la logique et la politique avec Louis Couturat (1897–1913)*. Éditions Kimé. (2001)
39. Thomason, R. (ed): *Formal Philosophy: Selected Papers of Richard Montague*. Yale University Press. (1974)
40. Wiener, N.: *Ex-Prodigy: My Childhood and Youth*. MIT Press. (1964)
41. Witkam, T.: *History and Heritage of the DLT (Distributed Language Translation) project*. Memorandum in MT Archive Info web site. (2006)