Du code au langage

Pierre Mounier-Kuhn

« Du code au langage », *Le Monde-Binaire*, 9 octobre 2015,
http://binaire.blog.lemonde.fr/2015/10/09/du-code-au-langage/

Du code au langage

Pierre Mounier-Kuhn[1]

Le progrès scientifique, la diffusion des sciences, reposent sur le langage. C'est pourquoi les mots ont tellement d'importance en sciences. C'est le cas aussi en informatique, une science naissante qui repose sur un vocabulaire forgé récemment et en évolution permanente. Un historien des sciences, Pierre Mounier-Kuhn revient pour Binaire *sur la genèse de mots essentiels comme* code *ou* langage.
Serge Abiteboul et Gilles Dowek.

À l'heure où le mot *code* envahit le discours politique, il est intéressant de revenir sur ce vieux terme et sur la manière dont il a peu à peu été remplacé, dans le vocabulaire des informaticiens, par le mot « *langage* ».

L'historien Mathias Dörries a noté que les scientifiques, quand une métaphore leur vient à l'esprit, ne se contentent pas de l'employer comme outil expressif : ils la poussent au bout de ses implications pour explorer toute sa valeur heuristique[2]. C'est précisément ce que l'on observe en étudiant l'émergence de la notion de *langage* en programmation. Je la résumerai d'abord en évoquant les étapes de cette évolution dans le petit milieu des proto-informaticiens français aux temps héroïques des ordinosaures, puis en nous portant sur la scène internationale.

Des calculatrices aux automates à programmes.

En novembre 1949, le fondateur d'une start-up abritée dans une ancienne usine automobile bombardée à Courbevoie, la Société d'électronique et d'automatisme (SEA), rédige un rapport interne qui constitue le premier projet d'ordinateur en France[3]. En résulte aussitôt un contrat de recherche avec le bureau des missiles de l'Armée de l'Air. Cette synthèse des réflexions de François-Henri Raymond et de son équipe décrit brièvement l'architecture d'un ordinateur. Elle donne le tableau des « codes d'ordre » de von Neumann prévus pour l'EDVAC développé à Philadelphie et la machine IAS de Princeton, et fournit un exemple d'application mathématique. Du point de vue de la programmation, elle ne parle que de *coding* et de *code*, défini comme « la suite des ordres précis telle qu'elle se trouve enregistrée en mémoire. »

---

1 Historien, CNRS et Université Paris-Sorbonne. Auteur de *L'Informatique en France de la seconde guerre mondiale au Plan Calcul. L'Émergence d'une science*, Presses de l'Université Paris-Sorbonne, 2010. Je remercie Gilles Dowek pour ses remarques sur ce texte.

2 M. Dörries, *Experimenting in Tongues: Studies in Science and Language*, Stanford University Press, 2002.

3 F.-H. Raymond, *Exposé sur la structure logique des grandes machines à calculer universelles*, 22 novembre 1949, rapport interne SEA.

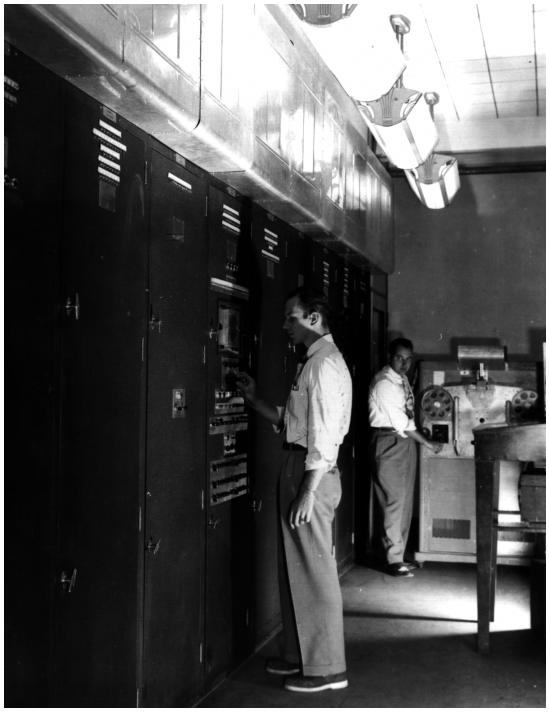

Figure 1. L'ordinateur EDVAC installé au centre d'essais militaire d'Aberdeen (Maryland), vers 1950.

Quelques années plus tard, en 1955, la SEA installe ses premiers ordinateurs chez des clients. Dans une série d'articles destinés aux ingénieurs, le chef d'entreprise explique leurs principes et leur fonctionnement, insistant sur leur nouveauté [4] : on doit les considérer non seulement comme des calculatrices, mais comme des systèmes conçus pour traiter des *programmes* ; non plus comme des machines, mais comme des *automates*. Raymond explique que le principe même de machine universelle fonctionnant en binaire implique « une *traduction* entre notre *langage* et le sien. » Et, filant la métaphore linguistique, il décrit le symbolisme permettant d'exprimer un processus de calcul comme « une convention d'écriture, qui constitue une règle de la grammaire du langage fonctionnel de la machine ». L'année suivante, dans un congrès à Milan, il parle à nouveau de ces « conventions de langage » qui permettent la « programmation automatique », laquelle revient à « un problème de traduction automatique ».

Tandis qu'IBM apporte Fortran à Paris en 1957, la SEA élabore son propre langage de programmation, PAF (Programmation Automatique des Formules), présenté en 1960. C'est d'emblée un langage conversationnel. La métaphore du langage et de la communication homme/ machine est poussée assez loin dans sa mise en œuvre technique, puisqu'il suffit de taper au clavier le début d'une formule pour que la machine la complète automatiquement : la « CAB 500 » donne à son utilisateur le sentiment qu'elle comprend même ses intentions !

4 F.-H. Raymond, « Les calculatrices numériques universelles », *Mémorial de l'Artillerie française*, 1955, n° 3 et 4. Une vingtaine d'articles et de livres sont publiés dans la foulée par la même équipe, afin de promouvoir les ordinateurs et la science nouvelle que Raymond appelle *L'Automatique des Informations*.

Figure 1. Ordinateur SEA CAB 2022 livré à Matra (1955).

Simultanément, le CNRS et l'Armement créent un Centre d'études de la traduction automatique, où se croisent la programmation, la logique et le traitement automatique des langues. Tandis qu'un aventurier français de la recherche mathématique, Marcel-Paul Schützenberger, écrit avec le linguiste américain Noam Chomsky un article qui apporte un éclairage théorique à la programmation et jette les bases de la linguistique computationnelle[5]. Ce rapprochement est manifeste avec la création, dès 1963-1964, des premiers *Instituts universitaires de programmation*, à Paris et à Grenoble, où les théories du langage ont d'emblée leur place dans l'enseignement.

Une convergence d'intérêts

Comment l'informatique (ou, pour employer le vocabulaire de l'époque, le calcul électronique) est-elle ainsi passée du *codage* au *langage* ? Au-delà du cas français que je viens d'évoquer, un groupe d'historiens fédérés dans un projet européen s'est plongé dans les archives et dans les mémoires des pionniers pour comprendre ce processus. Processus décisif pour la construction de l'informatique elle-même, comme discipline et comme activité professionnelle. Et qui s'observe d'abord dans les pays pionniers : l'Angleterre et les États-Unis, puis l'Allemagne[6].

L'idée de machines avec lesquelles on puisse communiquer s'est répandue après la guerre. Notamment avec l'ouvrage de Norbert Wiener, *Cybernetics, or Control and Communication in the Animal and the Machine* (1948) et l'article d'Alan Turing, paru dans *Mind* en 1950, dont le fameux test décrivait un dialogue en langage naturel avec un être humain et un ordinateur. Si l'on pouvait construire des « cerveaux électroniques », le langage ne devait-il pas être un de leurs attributs ? Ces réflexions cybernétiques ouvraient des perspectives inédites, mais leur anthropomorphisme allait rapidement les éloigner des préoccupations des praticiens.

---

5 N. Chomsky et M.-P. Schützenberger, « The algebraic theory of context-free languages », dans P. Braffort et D. Hirschberg (éd.), *Computer Programming and Formal Systems*, North Holland, 1963, p. 118-161.

6 Le projet européen *Soft-EU* a rassemblé pendant quatre ans une douzaine d'historiens venus des deux côtés de l'Atlantique. Je résume ici l'article de D. Nofre, M. Priestley et G. Alberts, « When Technology Became Language: the Origins of the Linguistic Conception of Computer Programming », *Technology and Culture*, 55, 1 (2014):40-75.

Si dès 1947 apparaît aux États-Unis le terme « langage machine » pour désigner le code binaire traité par les circuits électriques des grands calculateurs, c'est le mot *code* ou *codage* qui domine. Et pendant des années chaque ordinateur aura son système de codage particulier, développé localement par son équipe de programmeurs. Ce qui ne présente guère d'inconvénients tant que ces machines sont peu nombreuses – il en existe au plus 200 dans le monde en 1955. Les améliorations consistent à développer des *autocodes*, déchargeant le programmeur de tâches routinières comme la traduction de formules algébriques en langage machine[7]. Le résultat le plus abouti est la première version de ForTran, présentée en 1954 sur l'IBM 704.

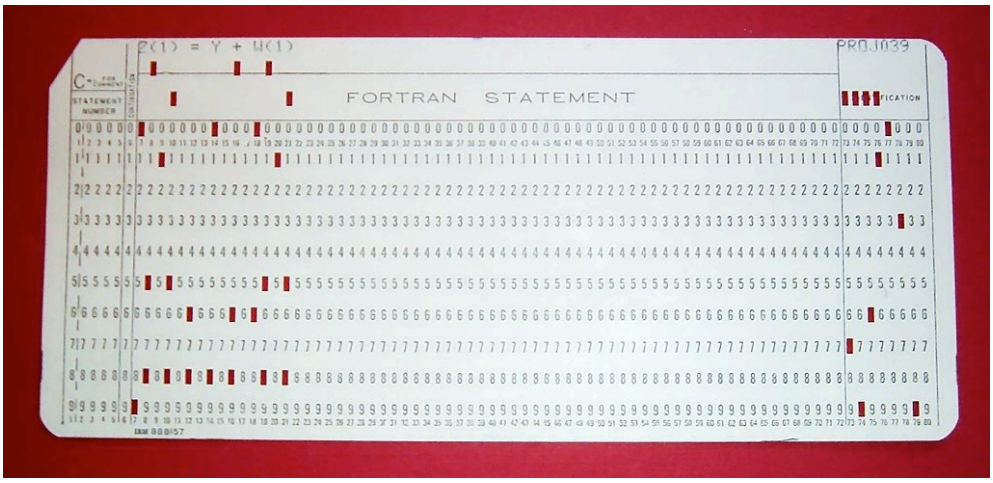

Figure 2. Carte perforée contenant une instruction d'un programme Fortran.

Les changements vont être impulsés par une convergence d'intérêts et de projets très divers. Le premier est d'ordre économique, lié à la croissance rapide du parc d'ordinateurs. Dès le milieu des années cinquante, les plus gros centres de calcul, dans l'armement ou l'aéronautique, possèdent plusieurs ordinateurs de modèles différents ou commencent à remplacer leurs premiers « cerveaux électroniques » par des machines de série. Ils découvrent le coût grandissant de la re-programmation : on doit non seulement réécrire chaque programme, mais aussi former de nouveaux programmeurs aux particularités de chaque architecture, dans une tension forte entre la rareté des compétences et la pression exercée sur la R & D par les besoins de la guerre froide.

Ces grands utilisateurs commencent donc à imaginer des techniques de programmation communes. IBM soutient cette tendance, comprenant que l'inflation incontrôlée des charges du software dissuaderait les clients d'acheter des matériels ; la mise au point de Fortran est un élément de solution. En mai 1954, lors d'un symposium sur les techniques de programmation automatique organisé par l'Office of Naval Research, plusieurs mathématiciens (Saul Gorn, John Carr, etc.) exposent leurs expériences : élaborer un « code universel » ou un « *universal computer language* », appuyé sur des méthodes de conception utilisant des schémas normalisés (ordinogrammes ou *flowcharts*), indépendants des modèles particuliers de machines, permettrait à la fois de maîtriser les coûts, d'améliorer la lisibilité et la qualité des programmes. Et de proposer aux étudiants des cours de programmation qui ne se limiteraient pas à des « recettes de cuisine ».

S'y ajoute une motivation politique. Ces universitaires voient dans un futur langage universel (« commun », précise-t-on souvent) un moyen d'échapper à l'emprise grandissante des constructeurs d'ordinateurs. Et de conserver la libre circulation des idées au sein des communautés de chercheurs, dans l'Université comme dans l'Armement.

---

7 Maurice V. Wilkes, Stanley Gill, David Wheeler, *The Preparation of Programs for an Electronic Digital Computer*, Cambridge, Addison-Wesley, 1951). Cet ouvrage anglais est le premier traité de programmation.

En 1955 un autre mouvement s'amorce dans le même sens : les clubs d'utilisateurs. Les utilisateurs commencent à s'organiser en groupes spécialisés par type d'ordinateurs. Leur principale activité consiste à échanger, à mettre en commun des programmes et des techniques de programmation, à établir des standards qui transcendent les particularismes locaux[8]. La tâche se complique d'autant que commencent à se multiplier les langages adaptés à chaque famille d'application.

C'est dans ce contexte que sont initiés quatre projets. Inspirée par les travaux d'Alan Perlis sur la compilation, IBM s'appuie sur son groupe d'utilisateurs SHARE pour rendre Fortran utilisable sur ses différents calculateurs scientifiques. Le groupe d'utilisateurs d'Univac, USE, imagine un supra-langage qui faciliterait l'échange de bibliothèques de programmes, et qui inspire un projet similaire de l'US Army. Le Department of Defense réunit un comité pour définir un *Common business-oriented language* (Cobol). Enfin plusieurs organisations s'associent pour élaborer un « langage intermédiaire », entre les langages d'application et les codes machines ; ce *Universal computer-oriented language* (Uncol) serait plus économique, espère-t-on, que le développement d'un compilateur pour chaque langage d'application. Cette usine à gaz linguistique sera enterrée en 1961, tandis que Cobol et Fortran décolleront sur des trajectoires durables. Mais tous ces efforts installent progressivement l'idée que la programmation peut être pensée en elle-même, sans lien avec un matériel spécifique.

Entre temps, des sociétés savantes se sont attaquées au problème. L'Association for computing machinery (ACM) crée en 1957 un comité des langages. Elle est contactée par la société allemande de Mécanique et Mathématiques appliquées (GAMM), en vue de définir un langage scientifique commun. Démarche bien naturelle : les mathématiciens sont habitués à une représentation traditionnelle des mathématiques en termes linguistiques, comme « langage de la Nature », comme « grammaire de la Science », puis au XXe siècle comme un système symbolique formel. Rapidement le projet emprunte donc les chemins bien balisés de l'internationale des mathématiciens.

Ce projet Algol (Algorithmic Language) reflète aussi leurs valeurs : universalité, rigueur dans l'expression, définition *in abstracto* indépendamment des contingences matérielles. Il vise deux objectifs : être à la fois un langage utilisable sur tout ordinateur et un moyen rigoureux d'expression et de communication des algorithmes. Présenté en 1959 à Paris au premier congrès international de traitement de l'information, il est spécifié l'année suivante sous le nom Algol 60, et devient un véritable programme de recherches en même temps qu'un outil de programmation. Pour la première fois, un langage informatique est défini formellement : l'auteur du Fortran, John Backus, et le danois Peter Naur inventent une notation qui porte leurs noms (*Backus-Naur Form,* BNF), permettant de décrire les règles syntaxiques des langages.

Tous ces courants de pensée amènent progressivement à concevoir la programmation comme une activité en soi, autonome vis-à-vis du matériel informatique comme des mathématiques, ayant son identité technique, professionnelle et scientifique. La programmation devient une discipline scientifique. Cinq ans plus tard, deux des acteurs centraux de cette histoire, John Carr et Saul Gorn, décideront de créer un prix pour couronner d'éminents travaux dans cette spécialité en rendant hommage à la mémoire d'un pionnier tombé dans l'oubli : Alan Turing.

Pierre Mounier-Kuhn

---

8 P. Mounier-Kuhn, "Les clubs d'utilisateurs : entre syndicats de clients, instruments *marketing* et logiciel libre avant la lettre", *Entreprises et Histoire*, décembre 2010, vol. 60, n° 3, p. 158-169. http://www.cairn.info/resume.php?ID_ARTICLE=EH_060_0158.

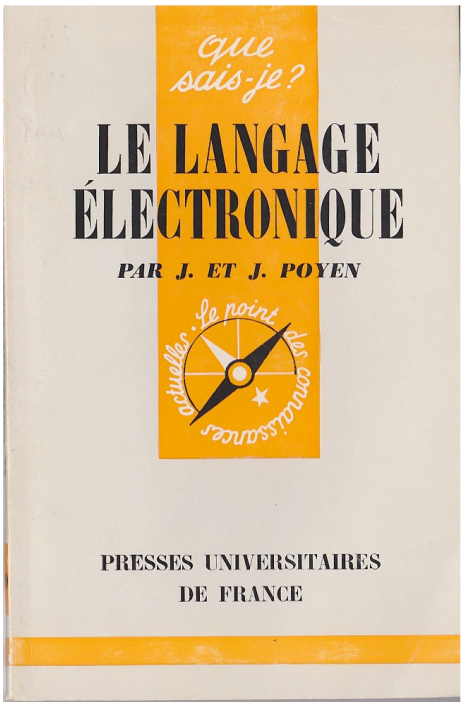

Figure 3. Jeanne et Jacques Poyen, *Le Langage électronique* (PUF, 1962).
Jeanne Poyen dirige le projet de compilateur Algol chez Bull. Jacques Poyen est chercheur au CNRS.

Pour aller plus loin :

[1] Pierre Mounier-Kuhn, *L'Informatique en France de la seconde guerre mondiale au Plan Calcul. L'Émergence d'une science*, Presses de l'Université Paris-Sorbonne, 2010.

[2] Le projet européen Soft-EU a rassemblé pendant quatre ans une douzaine d'historiens venus des deux côtés de l'Atlantique. Je résume ici l'article de D. Nofre, M. Priestley et G. Alberts, "When Technology Became Language: the Origins of the Linguistic Conception of Computer Programming", *Technology and Culture*, 55, 1 (2014):40-75.

[3] P. Mounier-Kuhn, « Les clubs d'utilisateurs : entre syndicats de clients, instruments marketing et logiciel libre avant la lettre", *Entreprises et Histoire*, décembre 2010, vol. 60, n° 3, p. 158-169. http://www.cairn.info/resume.php?ID_ARTICLE=EH_060_0158.