
Quelle place pour les utilisateurs-programmeurs dans une musée de l'informatique et de la société numérique ?

Proposition à partir de l'exemple du tableur

François-Marie Blondel, Éric Bruillard, Françoise Tort

*Laboratoire STEF, ENS Cachan / IFÉ (ENS Lyon)
61, avenue du Président Wilson F-94235 Cachan cedex, France
francois-marie.blondel@ens-cachan.fr
eric.bruillard@ens-cachan.fr
tort@ens-cachan.fr*

RÉSUMÉ. La programmation par des utilisateurs qui ne sont pas des informaticiens s'est répandue avec l'usage des tableurs dans les années 80. Cette activité a joué un rôle important dans le développement d'une culture informatique dans certaines catégories de la population, comptables, gestionnaires, financiers, techniciens, ingénieurs... Présenter l'activité des utilisateurs-programmeurs dans un musée à partir du tableur permet de discuter plusieurs aspects de la programmation (fonctions, macros, modèles, erreurs) sans exiger de connaissances techniques particulières. Des ouvrages, des documents de formation, des exemples de feuilles de calcul et de tableurs ainsi que des témoignages d'acteurs peuvent être recherchés ou produits pour alimenter des présentations dans un musée.

ABSTRACT. End-user programming has emerged with the spread of spreadsheet software in the 80's. This activity played a significant role in the development of computer knowledge within professionals like accountants, managers, technicians, engineers... Using spreadsheets to represent end-user programming in a museum can be an opportunity to discuss issues about programming (functions, macros, models, and errors) without requiring specific technical background. Resources for showings can be found in books, technical documentation, training courses, spreadsheets and interviews of typical users.

MOTS-CLÉS : utilisateurs, programmation, tableur.

KEYWORDS: end-user programming, end-user development, spreadsheet.

1. Introduction

Travaillant sur les questions d'éducation et de formation à l'informatique, nous nous sommes intéressés depuis plusieurs années à une catégorie d'utilisateurs assez peu étudiée en France, celle des utilisateurs d'instruments informatiques qui conçoivent ou développent des programmes et qui pour autant ne sont pas des informaticiens, au sens où ils ne se revendiquent pas comme des professionnels de l'informatique.

Ces utilisateurs particuliers sont apparus en nombre avec l'arrivée de la micro-informatique et le développement de l'usage des tableurs dans les années 80 (Bruillard et Blondel, 2007). À côté des amateurs passionnés assez nombreux à cette époque, est apparue une catégorie d'amateurs professionnels, c'est-à-dire d'utilisateurs qui ne sont pas des informaticiens mais qui programment et développent pour des raisons liées principalement à leur activité professionnelle.

Durant les trente dernières années, l'industrie des services s'est fortement développée, les métiers de l'informatique se sont spécialisés, la formation des informaticiens a connu un essor important. Même si elle n'est pas l'objet d'un recensement particulier, il semble que la proportion des utilisateurs programmeurs n'ait pas vraiment diminué, montrant ainsi l'importance de ces acteurs dans l'éventail des usagers de l'informatique.

Peut-on rendre compte de ces usages et de ces usagers dans un musée de l'informatique ? Quels seraient les objets et quelles activités les plus caractéristiques de ces usages ? En quoi le tableur constitue un exemple à la fois dans ces aspects historiques et plus contemporains ? C'est ce que nous souhaitons aborder dans cette communication.

Dans cette proposition, nous préciserons ce que sont ces utilisateurs, en quoi le tableur est un objet intéressant pour un musée de l'informatique et de la société numérique et nous évoquerons quelques propositions de ressources susceptibles d'alimenter des présentations.

2. Des utilisateurs qui programment

En prenant le point de vue de l'emploi des instruments logiciels, nous distinguerons pour simplifier trois catégories d'utilisateurs : ceux qui s'intéressent uniquement aux résultats qu'ils peuvent produire avec ces instruments, ceux qui s'intéressent aussi à l'adaptation de ces instruments à leurs besoins et ceux qui s'intéressent à la conception de ces instruments. Si la première catégorie, celles des « utilisateurs ordinaires », est assez bien identifiée, notamment pour des raisons marketing évidentes, si la troisième relève bien évidemment des professions de l'informatique, la deuxième catégorie, que nous appellerons les « utilisateurs programmeurs », est assez souvent négligée et parfois peu considérée.

2.1 Qui sont les utilisateurs programmeurs ?

S'intéressant à ces utilisateurs et à leurs activités, Bonnie Nardi (1993) précise que ce ne sont pas des « novices », que les ordinateurs sont pour eux des outils indispensables, qu'ils en font un usage important mais qu'ils ne souhaitent pas devenir des programmeurs professionnels.

Nous pourrions les caractériser comme des utilisateurs conduits à « programmer » sous une forme ou sous une autre, que ce soit pour réduire l'aspect répétitif de certaines tâches ou pour aboutir à des résultats qu'ils ne pourraient obtenir uniquement en modifiant des données ou en paramétrant les applications informatiques qui leur sont utiles. En ce sens, la finalité de leur activité de programmation est guidée par les résultats des traitements qu'ils ordonnent et non par les techniques employées pour les obtenir. Ce faisant, ils se placent eux-mêmes comme les premiers, et parfois les seuls, utilisateurs de leurs productions.

Cette activité de programmation par l'utilisateur final d'un produit est devenue importante avec la diffusion massive du tableur dans les années 80. La programmation par des amateurs, qui a connu un développement considérable avec la diffusion des micro-ordinateurs, est souvent restée dans le domaine privé. En revanche, le développement de calculs avec un tableur a été le fait de professionnels sur leur lieu de travail pour leurs besoins de suivi, de contrôle, de prévision, de modélisation, de prise de décision...

Depuis une quinzaine d'années, la programmation par des non informaticiens s'est développée chez d'autres utilisateurs comme les infographistes (écriture de scripts), les créateurs de sites web (gestion de bases de données), les gestionnaires de contenu, les enseignants (création de tutoriels ou de simulations).

Une population importante

Si le nombre des utilisateurs d'ordinateurs ou plus largement d'instruments informatiques peut être assez facilement estimé par des enquêtes nationales – ainsi dans l'enquête Credoc (2011) un tiers des actifs déclare avoir un usage quotidien d'internet et par conséquent d'ordinateurs sur son lieu de travail – il est plus difficile de savoir combien parmi ceux-ci ont une activité qui relève de la programmation.

La dernière étude cherchant à évaluer cette population a été menée par Scaffidi et al (2005). Ces auteurs estiment qu'en 2012, les Etats-Unis comptaient environ 3 millions de programmeurs professionnels, au moins 13 millions de personnes qui programment dans le cadre de leur activité professionnelle et plus de 50 millions d'utilisateurs de tableurs ou de bases de données. En faisant l'hypothèse que les proportions sont les mêmes pour la France, le nombre des utilisateurs programmeurs serait en France de l'ordre de 2,5 millions de personnes, soit environ le dixième de la population active en emploi. Cet ordre de grandeur paraît vraisemblable ; à titre de comparaison, Trémembert (2007) rapporte qu'en rapprochant les données de l'INSEE et du Credoc pour l'année 2005, près de la moitié (47%) des répondants qui

ont utilisé un ordinateur dans le dernier mois pour un usage professionnel déclarent avoir fait des calculs ou de la comptabilité.

2.2 Quelle culture informatique chez ces utilisateurs ?

La programmation des ordinateurs par des non-informaticiens a joué un rôle important dans le développement d'une culture informatique au moins dans certaines catégories de la population : comptables, gestionnaires, financiers, techniciens, ingénieurs....

À notre connaissance, une des études les plus complètes porte sur les usagers du tableur. Elle a été conduite en 2005 et 2006 par le Spreadsheet Engineering Research Project (SERP) de la Tuck School of Business du Dartmouth College, dans le but d'améliorer la conception et l'utilisation des feuilles de calcul au sein des entreprises¹. Les résultats du questionnaire rempli par près de 1600 répondants appartenant à des institutions différentes du monde des affaires et de l'informatique fournit une image très riche des pratiques des concepteurs et des utilisateurs de feuilles de calcul.

Plus de 80% des répondants considèrent que le tableur a un rôle très important, voire essentiel dans leur travail. Seulement 7% d'entre eux se considèrent comme débutants ou sans expérience, alors que 54% disent avoir une solide expérience et une certaine expertise et que près de 40% se disent très expérimentés et véritablement experts.

Ces chiffres sont à mettre en correspondance avec des travaux plus récents comme ceux de Dorn et Guzdial (2010) qui ont conduit des entretiens auprès d'une douzaine de concepteurs de sites web. Leur activité se répartit grosso modo à égalité entre la programmation (scripts, PHP) et l'infographie (Photoshop), avec de grandes disparités suivant les individus. Leur formation se fait sur le tas, puisqu'un seul d'entre eux déclare avoir obtenu un diplôme en informatique. Cependant la grande majorité est capable de reconnaître des concepts informatiques (variables, boucles...) même s'ils n'en maîtrisent pas toujours le sens.

Dans l'enquête du SERP de 2006, moins de 40% des utilisateurs déclarent avoir reçu une formation dans un cadre institutionnel et près de 18% n'ont pas reçu de formation du tout. En revanche, les exemples fournis par des collègues de travail ont joué un rôle pour plus de la moitié des utilisateurs. Comme l'avaient remarqué Nardi et Miller (1990) dans une étude ethnographique, il peut y avoir des formes de collaboration entre personnes ayant des niveaux d'expertise en programmation très différents. La formation de ces utilisateurs est un exemple caractéristique d'un apprentissage sur le tas dans lequel les modalités d'échanges entre pairs jouent un rôle important suivant les situations personnelles ou professionnelles. Dans le cas du tableur, les échanges entre utilisateurs sont encore extrêmement actifs comme en

¹ Les publications et les résultats de cette enquête sont disponibles à http://mba.tuck.dartmouth.edu/spreadsheet/product_pubs.html

témoigne par exemple le nombre de sujets en cours de discussion sur un site d'échanges sur la programmation².

Si le tableur n'est plus le seul instrument des utilisateurs programmeurs, les pratiques ont-elles pour autant beaucoup évolué ? Rien n'est moins certain. Ainsi, un examen du contenu des forums de discussion francophones sur la création de sites web laisse penser que nombre de participants sont encore des utilisateurs qui se forment sur le tas.

L'émergence d'une activité de programmation pour des besoins professionnels par des non-informaticiens a suscité un courant de recherche en interaction homme-machine (*end-user programming*) qui vise à comprendre en quoi consistent ces modes informels de développement de logiciel et qui cherche à concevoir des outils qui facilitent et améliorent ces développements. Centrées initialement sur la programmation, ces recherches prennent en compte plusieurs aspects du développement d'applications par les utilisateurs (*end user development*)³ et notamment les questions de génie logiciel (Ko *et al.*, 2011).

La permanence d'une activité de programmation et de développement par des non-informaticiens sur plus d'une trentaine d'années, la place qu'elle occupe dans les entreprises, le nombre des utilisateurs concernés nous paraissent peser significativement pour la prise en compte de cette activité dans un musée de l'informatique et de la société numérique.

3. Le tableur et ses usagers

Comment rendre compte de cette activité dans un musée ? Quels objets et quels utilisateurs présenter ? A l'évidence, le tableur paraît incontournable. Non seulement, ce produit est à l'origine d'un courant de programmation sur le lieu de travail par des non informaticiens, mais il en reste probablement l'exemple le plus répandu à l'heure actuelle.

La majeure partie des ordinateurs de bureau, des ordinateurs portables ou encore des tablettes sont actuellement commercialisés avec une suite bureautique qui incorpore systématiquement un tableur. Sa présence sur le marché ne semble pas remise en cause si l'on se réfère à l'offre commerciale et à l'offre issue du logiciel libre. Ainsi l'entrée correspondante de Wikipedia en langue anglaise⁴ ne cite-t-elle pas moins d'une cinquantaine d'offres de tableurs, qu'ils soient en ligne, indépendants ou inclus dans une suite bureautique.

² On pouvait dénombrer pas moins de 10 000 fils de discussion sous la rubrique « Excel » sur le forum Developpez.com en septembre 2012.

³ On pourra notamment consulter la définition qu'en donnent Burnett et Scaffidi (2011) dans une encyclopédie en ligne sur l'interaction homme-machine.

⁴ http://en.wikipedia.org/wiki/List_of_spreadsheets. Consulté en sept. 2012

3.1 La prise en compte des utilisateurs dans la conception

Les premiers tableurs pour micro-ordinateurs ont été conçus par des « inventeurs » qui ne se définissent pas seulement comme des informaticiens mais aussi comme des utilisateurs. Le point de vue de l'utilisateur est primordial pour les concepteurs comme le signale Bob Frankston sur son site⁵ :

« Since we were used to working with powerful computers without worry about the clock running, we already had the experience of focusing on the users needs rather than the computers needs. »

Le rôle des utilisateurs dans la diffusion du tableur a été primordial ; l'engouement des utilisateurs a été tel que le tableur est souvent considéré comme une des premières applications majeures (*killer application*), désignant ainsi un produit logiciel qui à lui seul justifie l'acquisition d'un ordinateur (Power 2004).

Le tableur est un instrument malléable qui a pu s'adapter, parfois trop facilement comme on le verra plus loin, aux intentions et aux habitudes de disposition des données et des calculs par les utilisateurs. A tel point que les évolutions qui ont été proposées pour répondre aux exigences d'une programmation plus conforme aux canons du génie logiciel ont été vouées à l'échec, du moins d'un point de vue commercial. On pourrait citer ici le point de vue exprimé par Adam Bosworth⁶ à ICSOC04 en novembre 2004, à propos de l'échec d'une des plus belles tentatives de « normaliser » le tableur, *Lotus Improv* :

« Consider the spreadsheet. It is a protean, sloppy, plastic, flexible medium that is, ironically, the despair of all accountants and auditors because it is virtually impossible to reliably understand a truly complex and rich spreadsheet. Lotus corporation (now IBM), filled with Harvard MBA's and PhD's in CS from MIT, built Improv. Improv set out "to fix all this". It was an auditors dream. It provided rarefied heights of abstraction, formalisms for rows and columns, and in short was truly comprehensible. It failed utterly, not because it failed in its ambitions but because it succeeded. »

3.2 Une programmation d'un genre particulier

Rappelons brièvement ce qu'est la programmation d'un tableur au sens le plus élémentaire. Une feuille de calcul est composée de *cellules* disposées sur un tableau en lignes et en colonnes ; chaque cellule peut contenir soit une *valeur* introduite par l'utilisateur, soit une *formule*, en pratique une fonction, qui prend ses valeurs dans des cellules de la feuille ou d'une autre feuille et retourne son résultat comme valeur de la cellule où elle a été introduite. Toute modification de la valeur d'une cellule entraîne le recalcul des formules des cellules qui en dépendent. La conception d'un *programme* fait intervenir la création des cellules sur les feuilles de calcul et la

⁵ <http://www.frankston.com/public/?name=ImplementingVisiCalc>

⁶ <http://adambosworth.wordpress.com/2004/11/18/iscoc04-talk/>

rédaction des formules qui aura pour conséquence d'établir les relations entre cellules et formules, suivant un modèle de calcul plus ou moins explicite.

La programmation d'un tableur se distingue par le fait qu'il n'y a pas de langage aux caractéristiques bien établies à l'origine mais plutôt un ensemble de fonctionnalités pour faciliter le développement rapide de calculs sur des tableaux (Abraham et al. 2009). À ce titre, le tableur n'est quasiment jamais employé pour l'enseignement de la programmation, même pour des débutants, alors qu'il est fréquemment présent dans la formation des étudiants en gestion⁷.

La programmation dans un tableur n'est pas *impérative* en ce sens que les instructions ne sont pas exécutées dans un ordre fixé par le programmeur. Alors qu'avec un langage impératif, le programmeur doit spécifier l'ordre d'évaluation des instructions, il en est complètement déchargé dans un tableur puisque les formules peuvent être ajoutées ou supprimées dans n'importe quelle cellule sans souci de l'ordre dans lequel elles seront évaluées. Cette absence de contrainte sur l'ordre des actions peut faire oublier à certains utilisateurs que l'écriture de formules relève bien de la programmation.

L'écriture de formules s'apparente à l'écriture de fonctions, à usage unique quand la formule n'apparaît que dans une seule cellule, ou à usage multiple en cas de recopie de la formule dans d'autres cellules, une manière de concevoir les itérations dans un calcul (Hodnigg et al. 2004). Pour étendre la liste des fonctions au-delà de celles qui sont fournies par le tableur, le programmeur devra faire appel à un langage externe associé au tableur, langage inspiré de Basic dans la plupart des cas. Faciliter l'écriture de fonctions par l'utilisateur a été sérieusement envisagé (Jones *et al.*, 2003).

Une autre caractéristique importante pour les utilisateurs est la possibilité de définir des *macros* permettant d'enregistrer et de nommer une suite d'actions, cette fonctionnalité étant très utilisée par les utilisateurs de tableurs pour remplacer des tâches répétitives par un *programme* dont ils n'ont pas à rédiger le code. On pourra noter que cette fonctionnalité est apparue très tôt – dès 1987, elle est présente dans la majorité des tableurs (Malloy, 1987) – et qu'elle est encore présente dans des applications très répandues comme les logiciels d'infographie par exemple.

Conçu pour pouvoir être adopté par des programmeurs initialement peu expérimentés, le modèle implicite de programmation est un modèle de développement incrémental. En effet, on peut n'écrire qu'une partie des formules sans pour autant rendre la feuille inutilisable, les seuls contrôles effectués portant sur la syntaxe des formules (impossible d'entrer une formule syntaxiquement incorrecte) ou sur une circularité dans les références (cette circularité pouvant entraîner une boucle infinie d'exécution peut être acceptée moyennant un paramétrage adéquat). Ainsi le tableur est parfaitement adapté à une méthode de

⁷ Parmi les exceptions, on pourra noter l'ouvrage de Soria et al. (1998) qui propose une initiation à la programmation à partir de Word et Excel.

tests et d'essais successifs d'hypothèses de présentation des données et de calculs, ce qui a fait son succès auprès des analystes⁸.

Développer un programme par essais et erreurs n'est pas une pratique particulièrement recommandée en programmation mais elle peut être revendiquée en tant que telle par des spécialistes en gestion comme T. Grossman (2002) :

Spreadsheet users sometimes start programming with only a vague idea of what they are doing. Software engineering wisdom suggests this is poor programming practice. However, this can be excellent business practice, because exploratory modeling teaches users about their business.

Les tentatives pour adapter la conception d'une application aux évolutions changeantes des besoins exprimés se retrouvent dans des méthodes qui se revendiquent itératives et adaptatives comme le développement rapide d'applications ou les méthodes agiles.

Mais ce qui rend plus facile un développement incrémental de calculs sur un tableur en constitue aussi un des défauts majeurs.

3.3 Heurs et malheurs du tableur

Le développement de feuilles de calcul complexes par des non-professionnels de la programmation recèle des risques pour les entreprises. Ceux-ci ont été identifiés et analysés assez tôt ; ainsi, dès 1987, Brown et Gould publient une première analyse des erreurs que l'on pouvait trouver dans les feuilles de calcul à cette époque. Robert Panko (2000) qui a longtemps étudié ce phénomène estime que la proportion de tableurs contenant des erreurs oscille entre 20% et 40%. D'où une abondante littérature sur l'analyse des erreurs, sur les méthodes d'audit et sur les outils pour les détecter ou les éviter. Cette question, toujours d'actualité, est à l'origine d'un collectif, *European Spreadsheet Risks Interest Group* (EuSpRIG) qui organise une conférence annuelle et dont le site⁹ contient une page spéciale sur les erreurs contenues dans des tableurs dont les conséquences sont les plus remarquables.

Lors de la conférence EuSpRIG de 2007, plusieurs intervenants, dont Stephen Powell du Dartmouth College (SERP), rappellent que les erreurs sont encore très présentes et soulignent que la perte de temps est un problème pratique encore plus significatif que celui des erreurs. En analysant un échantillon représentatif de 25 feuilles de calcul, Powell et al. (2009) ont montré qu'un peu moins de la moitié des erreurs n'avaient pas d'impact quantitatif mais que quelques unes portaient sur des montants importants, pouvant aller jusqu'à une centaine de millions de dollars.

⁸ Berger M. (1982). Scenarios for Success: The Vision Of Spreadsheets. *Personal Computing*. n°4. Copie en ligne : <http://www.aresluna.org/attached/computerhistory/articles/spreadsheets/scenariosforsuccess>

⁹ <http://www.eusprig.org/>

Présenter de telles erreurs et en discuter les types et les causes possibles ne nécessite que très peu de connaissances, essentiellement sur la syntaxe de l'écriture des opérations et des références¹⁰.

4. Quelles ressources pour un musée ?

Parce qu'il en est à l'origine et qu'il en reste des traces nombreuses, passées et récentes, le choix du tableur pour représenter ce qu'est la programmation par les utilisateurs paraît s'imposer, à côté d'autres aspects plus contemporains liés aux développements du web ou de l'infographie. Nous nous centrerons donc essentiellement sur le tableur.

L'actualité du tableur et sa diffusion auprès d'un large public ont pour conséquence d'offrir un grand choix d'objets et de documents susceptibles de présentation dans un musée. De plus, les utilisateurs des premiers tableurs sont encore en mesure de témoigner des usages qu'ils en ont faits ou qu'ils en ont perçus. Nous discuterons ci-dessous quelques suggestions pour rechercher ou produire les éléments qui nous paraissent parmi les plus appropriés.

4.1 Les tableurs, les feuilles de calcul

Les principes qui sont à la base de la conception des tableurs ayant peu évolué depuis leurs origines, il serait intéressant de montrer plusieurs exemples de tableurs en fonctionnement sur plusieurs générations de machines. Les applications tableurs ont été diffusées dans de telles quantités qu'il ne devrait pas être difficile d'en choisir quelques unes parmi les plus représentatives.

A titre d'exemple, Dan Bricklin propose sur son site¹¹ une version de VisiCalc pour IBM PC. Ce programme, assez bien écrit pour fonctionner encore correctement sous Windows XP, utilise la documentation originale de VisiCalc fournie également par Bricklin.

Des exemples de feuilles de calcul ne manquent, ni dans la documentation distribuée avec les tableurs, ni dans les ressources publiées pour l'éducation et la formation, ni sur la toile. À partir d'exemples simples comportant des défauts, on pourrait aussi aborder le statut de ce que l'on nomme souvent « erreurs » dans un programme, leurs causes potentielles et quelques techniques employées pour les repérer et les corriger. Par ailleurs, montrer des exemples de très grandes feuilles de calcul permettrait d'exposer quelques questions relatives à l'identification de ces erreurs dans des programmes complexes mais cependant opérationnels : typologies d'erreurs, techniques d'audit.

¹⁰ Notons que la notation A3, C2... de style « bataille navale », adoptée dans VisiCalc pour sa simplicité est toujours en vigueur dans les tableurs actuels.

¹¹ <http://www.bricklin.com/history/vcexecutable.htm>

4.2 Documentation et publications

L'abondance de la documentation écrite relative au tableur donne une idée de la variété des publics auxquels il s'adresse. Ainsi une requête effectuée sur le site de vente en ligne *Amazon.fr* en septembre 2012 et comportant le seul terme « Excel » retourne pas moins de 2500 ouvrages dont plus d'une cinquantaine publiés en 2012 ! La même requête effectuée sur le catalogue des bibliothèques universitaires retourne environ 1500 ouvrages en langue française publiés entre 1986 et 2012. On pourra au moins en déduire que l'utilisation du tableur a été une préoccupation largement partagée dans l'enseignement supérieur.

Les documents d'enseignement et de formation sont révélateurs de la manière dont la programmation est envisagée. A défaut d'une analyse approfondie des ces documents, on pourra cependant remarquer que les exemples de calcul sont très souvent présentés comme études de cas, inspirés de situations professionnelles. A titre d'exemple, on pourra noter que l'épreuve d'informatique du diplôme d'expert comptable comporte pour plus de la moitié des cas un travail pratique sur tableur¹². Trouver des exemples représentatifs de professions variées ne devrait pas poser de difficultés particulières.

La presse micro-informatique grand public a consacré de nombreux articles aux tableurs et à leur utilisation. Pour le monde anglophone, le magazine *Byte*, revue de référence en micro-informatique des années 80 et 90, contient des articles qui discutent du succès des tableurs et les chroniqueurs de l'époque présentent des arguments intéressants pour expliquer les raisons de ces succès, notamment sur l'interface et la prise en compte de l'utilisateur¹³. On trouvera des articles analogues dans la presse francophone, *l'Ordinateur Individuel*, *Soft&Micro*, *Science & Vie Micro*, *Pom's*, ...

Le tableur a rapidement occupé une bonne place dans la presse spécialisée. A titre d'exemple, dans l'un des tous premiers articles consacrés à l'ordinateur individuel¹⁴ dans l'encyclopédie *Les techniques de l'ingénieur*, le tableur arrive en tête des trois applications développées dans l'article, avant le traitement de texte ! Mais il n'a pas pour autant quitté l'actualité ; un billet publié sur *Le Monde Informatique* en juillet 2012 discute de l'intérêt relatif du tableur parmi les outils pour la décision : « La BI 4.0 tirillée entre le Big Data et Excel »¹⁵.

4.3 Les témoignages d'acteurs

Les déclarations publiques des « inventeurs » du tableur et des commentateurs de cette époque ne manquent pas (Bruillard et Blondel, 2007). Les témoins

¹² Avenel M., Reymann S., (1999). Mathématiques appliquées, Informatique – épreuve n°5 DECF, Paris : Foucher. 368 p.

¹³ Voir par exemple : *Byte*, vol. 8, N° 12, dec 1983, dossier *Easy Software*.

¹⁴ *Ordinateur Individuel*, article H 170, 1984.

¹⁵ <http://www.lemondeinformatique.fr/actualites/lmi/lire-la-bi-40-tiraillee-entre-le-big-data-et-excel-49774.html>

francophones de cette période sont beaucoup moins nombreux à s'être exprimés. Il faut cependant mentionner quelques exceptions notables et particulièrement les publications d'Hervé Thiriez, professeur de modélisation à HEC, auteur d'une douzaine ouvrages sur les tableurs, de VisiCalc à Javelin, créateur de lettres spécialisées et de revues et qui publie un blog intitulé « Monsieur Excel - Pour tout savoir faire sur Excel ! »¹⁶.

Recueillir les témoignages des utilisateurs français des tableurs dans leur contexte professionnel serait une contribution importante. Les quelques entretiens que nous avons pu mener en 2005 auprès d'utilisateurs représentatifs inciteraient à poursuivre ces investigations. On trouvera quelques extraits de ces entretiens en Annexe I.

Discussion

Le tableur, objet facilement identifiable par un large public depuis quelques décennies, présente de nombreux atouts pour une présentation de la programmation par les utilisateurs. D'un point de vue historique, il permet de montrer comment un instrument informatique peut notablement se transformer, en particulier son interface, alors que les principes à la base de sa conception, et les techniques employées pour le programmer n'ont quasiment pas changé depuis les origines.

Il serait intéressant de mettre en relation, dans un musée, le rôle du tableur dans la panoplie des techniques de calcul et notamment de celles qui font appel à des représentations en tableau (Campbell-Kelly et al. 2003).

Ce modèle de tableau a une grande importance dans notre culture écrite, comme le souligne Goody (1977) en montrant le rôle qu'ont joué les listes et leur organisation spatiale dans la différenciation qui s'est opérée entre la langue orale et la langue écrite. Le tableur est un moyen d'informatiser ces tableaux, ce qui, au-delà des utilisations comptables, lui confère une importance qui nous conduit à interroger sa place actuelle dans l'éducation et la formation.

Au-delà du tableur, d'autres exemples de programmation par les utilisateurs peuvent avoir leur place dans un musée. Le développement de sites web, qui rassemble aussi bien des professionnels de l'informatique que des spécialistes d'autres disciplines ou des amateurs, est une activité comportant souvent une part de programmation. Exposer cette activité dans un musée permet d'évoquer des questions de langages, de normalisation, de développement collectif, de partage de ressources... et de formation.

Bibliographie

Abraham, R., Burnett, M., & Erwig, M. (2009). Spreadsheet programming. *Wiley Encyclopedia of Computer Science and Engineering*. <http://onlinelibrary.wiley.com/doi/10.1002/9780470050118.ecse415/full>

¹⁶ <http://monsieur-excel.blogspot.fr/>

- Brown P. S., Gould J. D. (1987). An experimental study of people creating spreadsheets. *ACM Transactions on Information Systems*, vol. 5, n° 3, p. 258–272.
- Bruillard, E., Blondel, F.-M. & Tort F. (2008). *Projet DidaTab, Didactique du tableur*. Rapport final. STEF. http://www.stef.ens-cachan.fr/docs/rapport_didatab_2008.pdf
- Bruillard É., Blondel F.-M. (2007). *Histoire de la construction de l'objet tableur*. Pré-publication. hal-00180912, <http://hal.archives-ouvertes.fr/hal-00180912/fr/>
- Burnett, M. M., C. Scaffidi, (2011). End-User Development, In: Soegaard, M. and Dam, R. F. (eds.), *Encyclopedia of Human-Computer Interaction*, Aarhus, Denmark : The Interaction-Design Foundation, http://www.interaction-design.org/encyclopedia/end-user_development.html
- Campbell-Kelly M., Croarken M., Flood R., and Robson E. (eds.) (2003). *The history of mathematical tables: from Sumer to spreadsheets*, Oxford University Press, 376 p.
- Dorn, B., & Guzdial, M. (2010). Learning on the job: Characterizing the programming knowledge and learning strategies of web designers. *Proceedings of the 28th international conference on Human factors in computing systems*. p. 703–712. <http://dl.acm.org/citation.cfm?id=1753430>
- Goody J. (1977). *The domestication of the savage mind*, Cambridge University Press. Trad. française, *La raison graphique, la domestication de la pensée sauvage*, Les éditions de Minuit, 1979.
- Grossman T. A. (2002). Spreadsheet Engineering: A Research Framework, *Proceedings of the European Spreadsheet Risks Interest Group Symposium*, Cardiff, Wales, <http://arxiv.org/ftp/arxiv/papers/0711/0711.0538.pdf>
- Hodnigg, K., Clermont, M., & Mittermeir, R. T. (2004). Computational models of spreadsheet development: Basis for educational approaches. *Proc. European Spreadsheet Risks Interest Group Conference*. <http://arxiv.org/abs/0801.4274>
- Jones, S. L. Peyton, Blackwell, A. and Burnett, Margaret M. (2003): A user-centred approach to functions in Excel. *SIGPLAN Notices*, vol. 38, n° 9, p. 165-176.
- Ko, A. J., et al. (2011). The state of the art in end-user software engineering. *ACM Computing Surveys*, vol. 43, n°3, p. 1-44.
- Malloy R. (1987). This new crop of advanced programs offers multidimensionality and natural language, *Byte*, vol. 12, n°7, p. 69-75.
- Nardi B. A., Miller J. R. (1990). An ethnographic study of distributed problem solving in spreadsheet development. *Proceedings of CSCW'90*, p. 197-208. <http://www.miramontes.com/writing/ethno-spreadsheet/>
- Nardi B. A. (1993). *A Small Matter of Programming: Perspectives on End User Computing*. Cambridge, MIT Press. 178 p.
- Panko R. (2000). Spreadsheet Errors: What We Know. What We Think We Can Do. *Proceedings of the Spreadsheet Risk Symposium; European Spreadsheet Risks Interest Group (EuSpRIG)*, Greenwich, England. http://panko.shidler.hawaii.edu/SSR/My papers/EUSPRIG_2000.htm
- Powell, S. G., Baker, K. R., & Lawson, B. (2009). Impact of errors in operational spreadsheets. *Decision Support Systems*, vol. 47, n°2, p. 126-132.

- Power, D. J. (2004). *A Brief History of Spreadsheets*, DSSResources.COM, World Wide Web, <http://dssresources.com/history/sshistory.html>.
- Scaffidi C., Shaw M., and Myers B. (2005). Estimating the Numbers of End Users and End User Programmers. *VL/HCC'05: Proceedings of the 2005 IEEE Symposium on Visual Languages and Human-Centric Computing*. http://www.cs.cmu.edu/~cscaffid/papers/eu_20050923_vlhcc.pdf
- Soria M. et al. (1998). *Initiation à la programmation par Word et Excel : principes et macros*. Paris, Vuibert informatique. 516 p.
- Tort, F. (2009). L'enseignement du tableur en France. *Informatique et progiciels en éducation et en formation*. Lyon, INRP, p.200-217.
- Trémembert J. (2007). *Mesure des usages de l'informatique et situations d'usage*, M@rsouin (Môle @rmoricain de Recherche sur la Société de l'information et les Usages d'INternet). <http://www.marsouin.org/spip.php?article159>.

Annexe. Extraits d'entretiens sur les principaux emplois du tableur par différentes professions

F.L. directeur administratif financier dans une entreprise multinationale de production de matériaux

« La première chose que je fais quand j'allume mon ordinateur, c'est d'un côté j'allume Outlook, parce qu'il faut que je voie tous les mails qui sont en train de tomber et quel est le travail que je vais avoir à faire dans la journée, [...] ; et automatiquement j'ouvre de toutes manières une feuille Excel parce que je vais à peu près y travailler entre huit et dix heures par jour. »

C.S. consultant senior dans un cabinet de conseil

« Excel ça s'utilise ... surtout pour les analystes, donc des débutants avec une expérience de un, deux ou trois ans ça s'utilise de moins en moins fréquemment parce que effectivement, c'est pour calculer et ... pas forcément pour présenter. Donc j'utilise moins en moins Excel. »

« J'utilise Excel ... pour des calculs ... aussi comme base de données de temps en temps, parce que c'est simple, parce que c'est rapide. J'utilise Excel de temps en temps pour faire des graphiques, mais normalement les graphiques sont après transformés en PowerPoint... »

F.P. ingénieur d'études dans une administration publique

« On a de toutes façons des tenues très administratives ... de courrier, qui se font très souvent sous Excel. C'est pas qu'on ait une base de données de courriers trop importante, ça, ça peut se faire sous Excel. »

« On assure aussi des petits développements en utilisant les capacités d'équation de Excel... des petits développements calculatoires. »

« On va avoir un gros module de calcul, qui va nous donner les bases de calculs, et ensuite ces paramètres on va les réinjecter dans une base ..., dans une feuille de calcul Excel pour avoir des résultats très très pointus sur une ou deux petites particularités de la structure. »

« Il y a une troisième utilisation d'Excel qu'on faite, ... c'est l'exploitation des résultats de logiciels un peu plus complexes ... C'est-à-dire que le gros logiciel qu'on utilise pour faire le calcul des structures nous donnait autrefois des listings 'draps de lit', maintenant, il nous donne des fichiers qu'on importe dans Excel, dont on extrait par des macros uniquement les résultats qui nous intéressent pour faire de l'exploitation digestible type PowerPoint, ou des feuilles graphiques ... Alors on pourrait utiliser les capacités graphiques des programmes de départ, mais en général on peut avoir une personnalisation plus importante sous Excel qu'avec un programme général. »

H.O. contrôleur de gestion dans une entreprise multinationale de production de matériaux

« Je suis au contrôle de gestion donc je passe énormément de temps dans les chiffres et en particulier dans Excel. »

« Alors en fait Excel on s'en sert, ... je dirais comme outil de travail pour faire, je dirais de but en blanc, des tableaux d'analyse, avec ... les différents onglets, les formules de calcul, ... beaucoup les fonctions : j'utilise beaucoup les fonctions... un petit peu les graphiques, peut-être pas autant qu'on devrait les utiliser mais disons que ça, c'est les fonctions principales qu'on utilise, ... c'est à la fois des documents de travail et des documents de présentation, des documents qu'on diffuse dans l'ensemble de l'activité ... de la branche, et on se sert aussi d'Excel ... des applications qui permettent de transférer ... de ces applications vers Excel. »

Biographies

François-Marie Blondel est ingénieur de recherche au laboratoire STEF. Après avoir travaillé à la conception d'environnements informatiques et de ressources destinés à l'enseignement des sciences, il consacre l'essentiel de son activité de recherche à l'étude des usages des TIC dans l'enseignement secondaire.

Éric Bruillard est directeur du laboratoire STEF et enseignant chercheur en informatique. Ses recherches sont consacrées à l'articulation entre la conception et les usages des instruments informatisés dans l'éducation et plus largement à la didactique de l'informatique.

François Tort est maître de conférences en informatique au laboratoire STEF. Ses recherches sont centrées sur l'enseignement de l'informatique à tous les niveaux d'éducation et de formation, avec une attention particulière portée à l'utilisation des instruments informatiques.