



HAL
open science

Histoire de la construction de l'objet tableur

Eric Bruillard, François-Marie Blondel

► **To cite this version:**

Eric Bruillard, François-Marie Blondel. Histoire de la construction de l'objet tableur. 2007. hal-00180912

HAL Id: hal-00180912

<https://hal.science/hal-00180912>

Preprint submitted on 22 Oct 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Histoire de la construction de l'objet tableur

Éric Bruillard et François-Marie Blondel
UMR STEF
ENS de Cachan - INRP, UniverSud

Version de pré-publication du 22 octobre 2007

Avant-propos

Si, en France, le mot *tableur* est complètement étranger à nombre de personnes, beaucoup d'entre elles connaissent *Excel*, comme nous avons pu le vérifier maintes fois, tant auprès d'enseignants que d'élèves, lors des nombreux entretiens effectués dans le cadre du projet DidaTab¹. Que l'usage ait transformé progressivement un nom de produit en nom commun atteste de la grande réussite de la société *Microsoft*, qui en a également imposé d'autres comme *PowerPoint* et *Word*. Cette confusion dans la manière de nommer les logiciels reflète une réalité, celle de la domination de ces produits sur le marché des logiciels pour micro-ordinateurs. Aussi n'apparaît-elle pas comme totalement abusive. Mais lorsqu'il s'agit de questions d'éducation, il semble qu'il faille y attacher quelque importance.

En effet, la maîtrise d'un produit commercial ne fait pas partie des objectifs habituels de la formation initiale en France. Ainsi confondre *tableur* et *Excel* peut conduire à renforcer certaines idées fausses. Notamment, en cantonnant le tableur dans un registre technique, cela peut laisser à penser que sa maîtrise ne pose pas de problèmes et donc qu'il n'est pas nécessaire de chercher à analyser les compétences nécessaires à cette maîtrise, encore moins les progressions permettant leur acquisition.

Mais préciser ces questions, notamment définir ce que peut être un tableur, n'est pas aussi simple. Les grands progiciels disposent de tant de fonctionnalités qu'on en utilise qu'une partie minime et chacun sait que personne ne peut les maîtriser complètement. Les versions successives apportent toujours des nouveautés et les interfaces sont régulièrement adaptées. Aussi déterminer quelle maîtrise est visée n'est pas aussi facile et nécessite de s'abstraire par rapport à des implémentations spécifiques du tableur. Un tel travail a été mené dans le projet DidaTab, analysant les fonctionnalités de base d'*Excel*, pris comme produit de référence et fixant de fait ce qu'est un tableur, afin de déterminer les compétences sous-jacentes à sa maîtrise.

Bien que notre perspective de recherche soit orientée vers l'éducation et la formation, nous allons adopter ici une approche différente, de nature historique, en essayant de déterminer comment s'est construit le tableur, et partant de cette analyse de tenter de caractériser ce que peut être un tableur. Différents éclairages techniques, économiques, juridiques, peuvent être privilégiés, et, en l'occurrence, la richesse est bien dans la multiplicité des points de vue complémentaires fournissant des éléments utiles. Nous allons essayer de les articuler, gardant la visée de mieux rendre compte de la construction de cet objet. Comment comprendre le tableur en 2007 en reprenant les différentes étapes qui en jalonnent l'histoire ?

¹ Le projet DidaTab a pour objectif d'étudier les usages du tableur chez les collégiens et les lycéens en mettant l'accent sur la manière dont se construisent ces usages au cours de la scolarité dans le secondaire.

Les principaux repères qui jalonnent cette histoire se situent d'abord autour de l'invention de *VisiCalc*, premier produit commercial qui a popularisé le tableur. Les termes eux-mêmes méritent une analyse et les concepteurs d'applications informatiques qui revendiquent cette appellation sont nombreux, les uns ayant développé des algorithmes et des programmes de traitement de tableaux comptables, antérieurement à *VisiCalc*, les autres ayant proposé des méthodes originales pour interagir avec des tableaux, qu'ils soient comptables ou non. Le succès commercial de *VisiCalc* a été suivi par un foisonnement d'initiatives, dont seules quelques unes ont marqué significativement les produits actuels. Suivre l'évolution de ces produits et de leurs usages permet de mieux cerner comment les concepteurs, les éditeurs de logiciels et les utilisateurs ont façonné les tableurs que nous connaissons. Analyser les utilisations actuelles du tableur et les caractériser aussi en termes de programmation conduit à s'interroger sur ce qui fait le succès et les limites du tableur. Enfin, observer l'émergence des tableurs en ligne invite à réviser la manière d'utiliser les tableurs en y introduisant une composante collective, entraînant par là une modification éventuelle de ces produits. Tels sont les principaux points que nous allons détailler dans les sections qui suivent.

I. Premiers repères pour raconter l'histoire du tableur

L'invention de *VisiCalc* est incontestablement le moment clé qui a conduit à la reconnaissance des tableurs en tant que catégorie particulière de logiciels. Campbell-Kelly dans son histoire de l'industrie du logiciel (2003), rappelle que *VisiCalc* a dominé le marché de 1979 à 1983, suivi de *Lotus 1-2-3* de 1984 à 1990, avant qu'*Excel* n'occupe à partir de 1991 la position dominante qu'il détient encore. Aussi peut-on facilement distinguer plusieurs périodes : les précurseurs que sont les logiciels de tableaux avant *VisiCalc*, l'invention même de *VisiCalc*, ses successeurs immédiats dans les années 80, et la domination progressive d'*Excel*.

Rendre compte de ces différentes phases s'est avéré beaucoup plus difficile qu'initialement prévu, du fait de manque de sources reconnues comme fiables, et de l'intensité des débats en cours entre précurseurs. Aussi en marge de la reconstruction de cette histoire, apparaissent des questions plus générales comme la brevetabilité des logiciels, aussi incidentes, comme le mode de fonctionnement de l'encyclopédie contributive Wikipedia, que nous allons brièvement ébaucher.

Auparavant, nous allons préciser comment cette analyse a pu être conduite et donner quelques repères lexicaux sur les termes utilisés pour désigner le tableur.

I.1. Les principales sources disponibles

Notre intention est de rassembler et d'analyser les principales sources disponibles et notamment les propos des acteurs qui ont eu – ou qui prétendent avoir eu – un rôle important dans la construction des tableurs, et de tenter d'en fournir une synthèse.

Comme bien souvent lorsqu'il est question d'informatique, de nombreux documents sont directement accessibles sur le web. Mais on s'aperçoit vite que les répétitions sont innombrables, que le nombre de sources réellement distinctes est assez réduit et que la fiabilité de certaines d'entre elles est à questionner.

Commençons par les documents les plus intéressants et les plus accessibles. Pour situer le tableur dans l'évolution du logiciel, on peut d'abord se référer aux ouvrages généraux consacrés à l'histoire du calcul, de l'informatique, de l'industrie du logiciel, donnant des repères intéressants, notamment le livre de Campbell-Kelly (2003).

Parmi les documents consacrés exclusivement au tableur et à son histoire, on peut citer les sources principales suivantes (au moins pour une première exploration de cette histoire) : un texte d'un consultant en systèmes d'information et de gestion, Dan Power (2004), le site de l'un des spécialistes du tableur, John Walkenbach², l'entrée *Spreadsheet* de l'encyclopédie Wikipedia en langue anglaise et enfin le site d'un des auteurs de *VisiCalc*, Dan Bricklin³. Le premier de ces documents donne les étapes essentielles et dans ses versions successives intègre des réactions des protagonistes, Bricklin et Frankston en particulier. Le second introduit les travaux précurseurs de Mattessich⁴. Le troisième, encyclopédique, fournit des pistes importantes pour identifier d'autres précurseurs et le dernier présente des détails précis et des commentaires argumentés sur l'invention de *VisiCalc*.

Pour la période plus récente qui commence dans les années 90, on notera qu'il n'existe pas, à notre connaissance, de document majeur relatant l'histoire d'*Excel*. John Walkenbach atteste du silence de Microsoft à cet endroit et, dans sa page de liens consacrée à l'histoire du tableur, commente ainsi un lien mort vers des informations sur l'histoire d'*Excel*⁵ :

« Interesting stuff from Microsoft. Well, it WAS interesting. Like many things on Microsoft's web site, this has disappeared. »

D'autres sites apportent des éclairages complémentaires, par leurs analyses ou par les documents primaires qu'ils proposent. Parmi ceux-ci, on peut citer Christopher Browne⁶ qui développe les questions actuelles relatives aux tableurs en les rapportant à des aspects historiques, Marcin Wichary qui a mis en ligne des articles de magazines spécialisés sur l'informatique personnelle des années 80 (Personal Computing, PC Magazine, Byte)⁷, le Computer History Museum⁸ qui a réalisé des entretiens avec des acteurs marquants de la même période, notamment celui de Mitch Kapor, à propos de *Lotus 1-2-3*, et enfin, en français, une synthèse de Michel Volle⁹.

I.2. Intermède lexical - origines du mot tableur

Tenant compte de l'origine nord-américaine des tableurs, c'est d'abord le mot anglais correspondant *spreadsheet* qu'il convient d'étudier. Celui-ci désigne une feuille de papier utilisée en comptabilité.

... Historically, a spreadsheet is an accounting ledger page that shows various quantitative information useful for managing a business ...

*Britannica online*¹⁰

Traditionally, accountants use a grid of rows and columns printed on special green paper to produce financial projects and reports. An electronic spreadsheet is software that simulates this accountant's paper pad or worksheet ...

*Macmillan Science Library: Computer Sciences*¹¹

Ou d'autres définitions équivalentes :

A spreadsheet is a rectangular table (or grid) of information, often financial information. (It is, therefore, a kind of matrix.) The word came from "spread" in its sense of a

² The Spreadsheet Page, <http://www.j-walk.com/ss/index.htm>

³ <http://www.bricklin.com/history/sai.htm>

⁴ <http://www.j-walk.com/ss/history/spreadsh.htm>

⁵ <http://www.j-walk.com/ss/history/index.htm>

⁶ Linux Spreadsheets : <http://www.cbbrowne.com/info/spreadsheets.html>

⁷ <http://www.aresluna.org/attached/computerhistory/spreadsheets>

⁸ <http://www.computerhistory.org/>

⁹ <http://www.volle.com/travaux/tableur.htm>

¹⁰ <http://www.britannica.com/eb/article-9396603/spreadsheet>

¹¹ <http://www.bookrags.com/Spreadsheet>

newspaper or magazine item (text and/or graphics) that covers two facing pages, extending across the center fold and treating the two pages as one large one. ...

*Wikipedia*¹²

Cela consacre la présence du tableur dans la lignée des instruments comptables, celui-ci apparaissant comme la traduction informatique de feuilles de calcul comptables, bien que le terme puisse apparaître dans d'autres contextes.

Un détour par le Québec peut être utile pour étudier l'éventail des traductions possibles du terme *spreadsheet*. Dans le grand dictionnaire terminologique de l'Office Québécois de la Langue Française¹³, le mot *chiffrier*, dans son sens comptable, représente :

- soit l'équivalent du mot anglais *work sheet*, avec pour définition :

Feuille ou tableau à colonnes multiples constitué de différentes sections où figurent respectivement la balance de vérification avant régularisations, les régularisations, la balance de vérification après régularisations et, dans les deux dernières sections, les éléments qui feront respectivement partie de l'état des résultats (ou compte de résultat) et du bilan.

- soit l'équivalent de *worksheet* (en un seul mot) :

Feuille de récapitulation où la balance de vérification avant la fermeture est redressée à la suite de l'examen des comptes afin de faciliter l'établissement des données qui figureront dans les états financiers.

Il correspond à *tableur* (ou *chiffrier électronique*) dans son sens informatique (*spreadsheet program*) :

Logiciel servant à la création et à la manipulation de tableaux numériques ou alphanumériques, utilisé à des fins de gestion et d'applications scientifiques.

Ce dictionnaire suggère les correspondants *tableur électronique*, *logiciel tableur* ou *gestionnaire de tableaux* et précise :

Il y a souvent confusion dans l'usage des termes *spreadsheet program* et *spreadsheet*. Le premier désigne le programme qui sert à effectuer des calculs automatiques (tableur). Quant à *spreadsheet*, il est utilisé à la fois comme synonyme de *spreadsheet program* et pour désigner le tableau affiché à l'écran dans lequel on entre des données (feuille de calcul). Employé dans le sens de « tableur », le terme feuille de calcul est inadéquat.

Au Québec, tableur devrait être utilisé de préférence à chiffrier, terme emprunté à la terminologie comptable québécoise et peu approprié à l'informatique.

Le terme gestionnaire de tableaux connaît un usage moins fréquent.

Quant au terme français « *tableur* », il s'agit d'un néologisme forgé par la commission de l'informatique et des composants électroniques, qui a été officiellement introduit par un arrêté du 30 décembre 1983¹⁴ avec pour définition :

Logiciel de création et de manipulation interactives de tableaux numériques.

Ni le terme ni sa définition ne portent le poids de la comptabilité ; *tableur* se rapporte à table ou à tableau et intègre l'idée d'une présentation tabulaire. Une piste qui peut aider à définir la notion même.

Le Jargon Français, dictionnaire informatique en ligne contributif, donne une définition de tableur un peu renouvelée¹⁵ :

¹² en.wikipedia.org/wiki/Spreadsheet

¹³ http://w3.granddictionnaire.com/BTML/FRA/r_Motclef/index800_1.asp

¹⁴ <http://www.criter.dglf.culture.gouv.fr/pls/DGPB/>, En tapant « tableur », on obtient la fiche INFO221 du 22/09/2000

Progiciel permettant d'effectuer des calculs répétitifs de façon massive, en rangeant données et formules dans un ou plusieurs tableaux (appelés « feuilles de calcul ») qui, en pratique, sont autant d'automates cellulaires. C'est aussi une manière intuitive de programmer. Exemple : Excel.

Si cette dernière définition n'est pas tout à fait exacte (sur les différences entre automates cellulaires et feuilles de calcul, voir par exemple l'analyse de Christopher Browne¹⁶), son intérêt est de faire le lien avec d'autres structures informatiques.

Après ce rapide tour d'horizon linguistique, nous allons essayer de déterminer les apports principaux du produit qui a popularisé le tableur : *VisiCalc*.

I.3. VisiCalc

La première source incontournable sur *VisiCalc* est celle produite par son concepteur Dan Bricklin qui en raconte l'histoire sur son site personnel¹⁷. Comme le rappelle Beaudoin-Lafon dans un article rétrospectif sur l'interaction homme-machine, l'idée première est une combinaison d'éléments d'interface et d'éléments de calcul. :

Bricklin, étudiant à Harvard, en a l'idée en utilisant une calculette Texas Instruments. Il imagine comment un système permettant la visualisation dite « tête haute » d'une feuille de calcul, piloté par un « trackball », lui faciliterait la résolution de ses exercices d'économie en lui permettant de tester rapidement plusieurs hypothèses.
(Beaudouin-Lafon, 2007)

A partir de cette intuition émise au printemps 1978, Bricklin a rapidement réalisé un prototype que Bob Frankston a repris. Ils ont créé ensemble la société *Software Arts* le 2 janvier 1979, et ont conclu un accord avec un éditeur, *Personal Software*, dirigé par Dan Fylstra. Bricklin expose la première étude de cas utilisant *VisiCalc* pour un travail d'étudiant sur une campagne marketing de Pepsi-Cola, en même temps qu'il conçoit les premières publicités.

Dans le récit que fait Frankston de la conception de *VisiCalc*¹⁸, il apparaît clairement que la prise en compte de l'utilisateur et plus précisément le souci de rendre le plus de choses visibles à l'interface est une préoccupation permanente des concepteurs. D'où ce commentaire de Ben Rosen, dans le *Morgan Stanley Electronics Letter*, du 11 juillet 1979¹⁹ :

“In minutes, people who have never used a computer are writing and using programs”.

En fait, cette phrase souligne l'impact majeur qu'a eu *VisiCalc* sur la démocratisation liée au développement de la micro-informatique. *VisiCalc* a été en partie responsable du succès de l'Apple II et a ainsi contribué à introduire les micro-ordinateurs dans les entreprises.

¹⁵ <http://jargonf.org/wiki/tableur>

¹⁶ Linux Spreadsheets : <http://www.cbbrowne.com/info/spreadsheets.html>

¹⁷ <http://www.bricklin.com/history/sai.htm>. Voir aussi une interview de Dan Bricklin, Bob Frankston et Dan Fylstra : *Three Minutes: Godfathers of the Spreadsheet*, <http://pcworld.about.com/news/Jun032004id116166.htm>

¹⁸ <http://www.frankston.com/public/?name=implementingVisicalc>

¹⁹ <http://www.bricklin.com/history/rosenletter.htm>

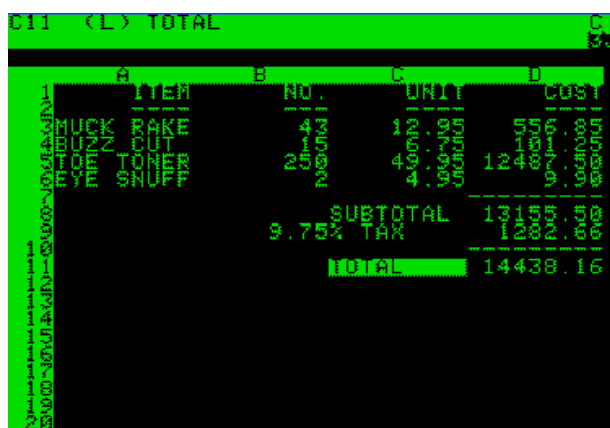


Figure 1. Une copie d'écran de VisiCalc sur Apple II²⁰

(Le vert est la couleur du moniteur de l'époque, ainsi que la couleur habituelle des *spread sheets*)

Pour Auray (2000), les nombreux détenteurs de diplômes de gestion arrivant sur le marché du travail avaient les compétences nécessaires pour utiliser un ordinateur et *VisiCalc* leur accordait l'indépendance.

« VisiCalc a permis d'émanciper les cadres de la tutelle de l'informatique classique » (Ichbiah, 1993, p. 122, cité par Auray p. 82).

Cela a conduit un public professionnel à la micro-informatique, créant ainsi un nouveau marché pour un produit resté jusqu'ici limité à des techniciens ou des amateurs. On peut considérer aussi que *VisiCalc* apporte une nouvelle manière de considérer un logiciel car il est « amusant » à utiliser. Il donne envie d'expérimenter.

« L'alchimie de VisiCalc tient au fait qu'il a introduit le ludique dans le contexte austère et guindé des professions d'encadrement » (Ichbiah, 1993, p. 123, cité par Auray, p. 171).

Selon Fertig²¹ (Campbell-Kelly, 2003, p. 235), la nature interactive, en temps réel, de *VisiCalc* n'était pas une simple amélioration des programmes de planification financière, mais constituait un nouveau paradigme. Le nom *VisiCalc*, raccourci de *Visible Calculator*, n'évoque en rien la comptabilité mais souligne, selon Frankston²², l'idée d'assistant pour le calcul (*smart calculator*).

Pour Piguet et Hügli (2004), dans leur chapitre dédié à l'histoire de la programmation,

« L'arrivée du tableur est importante car elle marque la familiarisation d'un large public avec une application non impérative » (p. 133).

Alors qu'avec un langage impératif, le programmeur doit spécifier l'ordre d'évaluation des instructions, il en est complètement déchargé dans *VisiCalc* 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.

Pour plus de détails sur la création de *VisiCalc*, on pourra également consulter les entretiens de Eric Bender avec Bricklin, Frankston et Flystra dans *Three Minutes: Godfathers of the*

20 Museum of User Interfaces, documents pour un cours de Ben Shneiderman, University of Maryland
<http://www.cs.umd.edu/class/spring2002/cmssc434-0101/MUIseum/applications/spreadsheethistory2.html>

21 *The Software Revolution: Trends, Players, Market Dynamics in Personal Computer Software*, North Holland, p. 178, 1985.

22 <http://dssresources.com/history/frankston4151999b.html>

*Spreadsheet*²³, le livre de Campbell-Kelly (2003, p. 234-236) ainsi qu'une histoire de la société *VisiCorp* par Dan Fylstra²⁴.

Les différents commentaires précédents nous permettent de donner quelques caractéristiques importantes de *VisiCalc* :

- C'est avant tout un outil de calcul interactif qui privilégie une présentation des données en tableau, plus qu'un outil comptable,
- C'est aussi un outil de programmation, mais accessible à des non spécialistes, conduisant certains auteurs anglophones à parler de *end-user programming*.

Cette capacité qu'offre *VisiCalc* à explorer rapidement des situations financières, fournissant directement un résultat, est aussitôt soulignée et tranchée par rapport aux réalisations informatiques antérieures. La possibilité de tester facilement des hypothèses (*what if analysis*) a très tôt marqué les observateurs :

Think of the number of times you have gone through an elaborate plan – one that involved numerical calculations – and worked the structure through to a conclusion. Upon presenting that result to superiors, you found the work destroyed by the simple question, “What if your assumptions aren't valid?”

With *VisiCalc*, this need be no cause for alarm. Your assumptions are represented by numbers on the electronic spread sheet. To change the assumptions, simply change the numbers. The program automatically calculates the new results that flow from the change. Many people say this ability to “what if” any kind of problem is *VisiCalc*'s greatest advantage.

What is this *VisiCalc*, anyway? *Personal Computing, issue 2, 1982, p. 154*²⁵.

Comme Bricklin l'avait prévu, le tableur devient un outil d'aide à la décision et non pas simplement un outil de calcul. En ce sens, *VisiCalc* a changé le rapport au micro-ordinateur dans le monde des affaires.

« Si, jusqu'alors, peu de gens avaient seulement entendu parler d'un « tableur », quelques années plus tard le terme ferait partie du vocabulaire des affaires, et travailler sur un tableur transformerait la vie professionnelle des cadres dans le monde occidental ». (Campbell-Kelly, 2003, p. 223).

II. Les précurseurs

VisiCalc a été une application majeure (*killer application*)²⁶ que l'on peut considérer comme représentative d'une période de l'histoire de la micro-informatique, mais ce produit original n'est pas apparu au milieu d'un désert logiciel. Des applications construites sur le principe de feuilles de calcul étaient disponibles dès les années 60 sur les ordinateurs fonctionnant en traitement par lots et plus tard sur les systèmes en temps partagé, mais leur manque d'interactivité en limitait l'utilité.

Où faut-il aller rechercher les précurseurs du tableur ? D'abord du côté de la comptabilité puisque c'est l'une des origines affirmées de ce produit. Mais le tableur est aussi un logiciel de calcul ; ses origines peuvent donc être recherchées dans l'histoire des moyens de calcul, et notamment celle des tables (*The history of mathematical tables: from Sumer to spreadsheets*, Campbell-Kelly *et al.*, 2003), ou de la programmation (*Du Zéro à l'ordinateur. Une brève*

²³ <http://pcworld.about.com/news/Jun032004id116166.htm>

²⁴ http://edesber.com/companies/vis_history.shtml

²⁵ <http://www.aresluna.org/attached/computerhistory/articles/spreadsheets/whatisthisvisicalcanyway>

²⁶ La traduction « application phare » proposée par Mounier-Kuhn, indique mal l'idée d'une application qui « tue » [les concurrents] sur le marché des ordinateurs.

histoire du calcul, Piguet et Hügli, 2004), ou des interfaces (*40 ans d'interaction homme-machine : points de repère et perspectives*, Beaudouin-Lafon, 2007).

Les débuts du calcul sur machine répondent à des besoins de résolution de questions comptables, ce qui donne une intrication difficile à démêler. Cet aspect n'a pas manqué d'intriguer et d'inspirer Jules Verne, dans son roman *Paris au XX^e siècle* redécouvert récemment :

« Michel se retourna et aperçut la machine n° 4. C'était un appareil à calculer.

Il y avait loin du temps où Pascal construisait un instrument de cette sorte, dont la conception parut si merveilleuse alors. [...] La maison Casmodage possédait de véritables chefs-d'œuvre ; ses instruments ressemblaient, en effet, à de vastes pianos ; en pressant les touches d'un clavier, on obtenait instantanément des totaux, des restes, des produits, des quotients, des règles de proportion, des calculs d'amortissement et d'intérêts composés pour des périodes infinies et à tous les taux possibles. Il y avait des notes hautes qui donnaient jusqu'à cent cinquante pour cent ! Rien de merveilleux comme ces machines qui eussent battu sans peine les Mondeux et les [?]

Seulement, il fallait savoir en jouer, et Michel dut prendre des leçons de doigté. »

(Jules Verne, *Paris au XX^e siècle*, Hachette, 1994, p. 68-69)

Il peut être intéressant de faire le lien entre cette description et un clavier imaginé par Douglas Engelbart, instrument permettant d'associer des combinaisons de touches à des commandes, mais qui n'a pas connu un grand succès, au contraire d'une autre invention de cet innovateur, la souris.

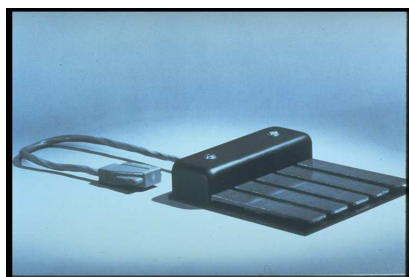


Figure 2. La production *keyset*²⁷ permettait de communiquer les commandes avec la main gauche alors que la souris était pilotée par la main droite.

Le tableur dans l'histoire de la comptabilité

C'est donc l'histoire de la comptabilité qui va permettre d'éclairer l'histoire du *spreadsheet*. Selon Cilloni et Marinoni (2006), les origines des tableurs sont à trouver dans les méthodes comptables développées dans la seconde moitié du XIX^e siècle, fondées sur l'algèbre matricielle, dans la tradition européenne de comptabilité mathématique. La littérature sur la genèse et le développement des tableurs est concernée par l'algèbre matricielle (voir Mattessich and Galassi, 2000 ; Mattessich, 2005 ; Cilloni, 2005 ; 2007).

Selon Mattessich (2005), de Morgan fut le premier à introduire les matrices en comptabilité en 1846. En 1889, Giovanni Rossi a proposé une douzaine d'exemples dans lesquels les matrices jouaient un rôle décisif. De plus, Rossi a suggéré que la matrice de comptabilité puisse être convertie en une machine à calculer sophistiquée, précurseur mécanique des tableurs électroniques. Ainsi le premier tableur connu serait le « *calcolatore tabulare meccanico automatico* », ou calculateur tabulaire mécanique automatique de Giovanni Rossi (1870), qui a permis une avancée décisive dans la relation entre algèbre matricielle et les matrices comptables (Cilloni et Marinoni, 2006 ; Cilloni, 2007).

²⁷ <http://www.bootstrap.org/chronicle/pix/pix.html#2B6>. History in pictures, Bootstrap Institute.

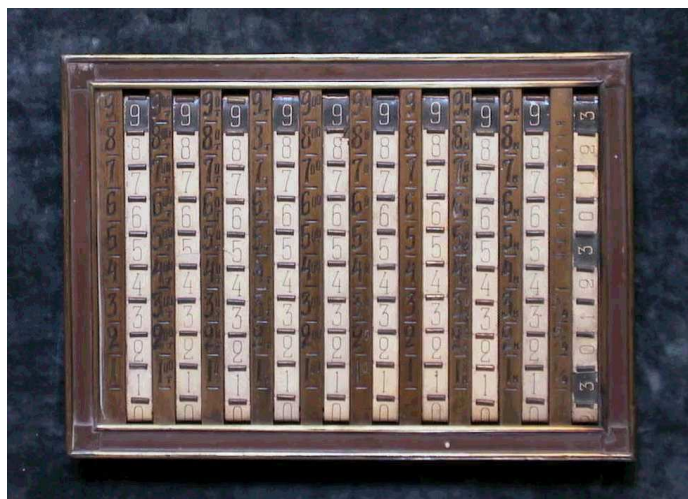


Figure 3. La machine à calculer de Rossi (extrait de Cilloni, 2007).

Ensuite, avec le développement des ordinateurs, la question s'est rapidement posée d'utiliser ces machines pour faire automatiquement des calculs comptables, notamment sous forme de tableau.

Mattessich a lui-même joué un rôle clé dans cette informatisation²⁸. Son nom apparaît dans la liste de soumission aux récompenses de l'American Accounting Association, pour les innovations dans l'enseignement de la comptabilité (Innovation in Accounting Education Award 1999)²⁹.

Mattessich souhaite qu'on lui reconnaisse la paternité de l'informatisation des tableaux comptables et revendique notamment l'idée de pouvoir placer une formule dans chaque cellule (2006, p. 28) :

“Among these ideas (of the 1961 paper and my subsequent simulation publications) the most important one is the use of mathematically formulated hypotheses behind the entries or individual cells in the matrix (spreadsheet) - in other words: an accounting entry need not consist of a numerical figure but may be an entire formula. Today - when these "formulas" appear either on the top or bottom of the spreadsheet screen - this idea seems to be quite self-evident. But in 1961 this was certainly not the case, as the resistance to my paper by some of my colleagues vividly illustrates.”

Notons que la comptabilité a largement évolué depuis les années 60, les ordinateurs et les tableurs n'étant pas étrangers à cette évolution.

Les auteurs d'histoires du tableur considèrent généralement que les travaux de Mattessich correspondent au premier tableur informatique. Mais d'autres noms apparaissent qui soulèvent des questions d'une autre nature. Cela va nous permettre d'ébaucher la question de brevetabilité des logiciels et de nous confronter au fonctionnement de l'encyclopédie Wikipedia.

²⁸ *Spreadsheet: Its First Computerization (1961-1964)* By Richard Mattessich, John Walkenbach
<http://www.j-walk.com/ss/history/spreadsh.htm>

²⁹ <http://aaahq.org/facdev/teaching/awardsub.htm>. Pioneering the Electronic Spreadsheet: Educational and Practical Implications as well as Analytical Background, Richard Mattessich,
<http://aaahq.org/facdev/teaching/submissions/mattessich.htm>. Il ne semble pas avoir obtenu de récompense.

La brevetabilité des logiciels et des algorithmes

VisiCalc n'a pas été breveté par son éditeur et Dan Bricklin explique³⁰ qu'à cette époque, les programmes informatiques étant considérés comme des algorithmes mathématiques et, à ce titre, vus comme des « lois de la nature », n'étaient pas brevetables. *Personal Software*, éditeur de *VisiCalc*, avait envisagé le dépôt de brevet, mais en avait été dissuadé par un juriste.

Dans une étude à caractère historique sur les brevets pour les logiciels, Campbell-Kelly (2005) rappelle que, après un premier jugement favorable en 1981, la jurisprudence en matière de brevets s'est peu à peu adaptée aux particularités des nouvelles technologies. Il estime que les brevets sont supérieurs aux autres solutions de protection de la propriété du logiciel (secret commercial ou copyright) du fait de l'ouverture qu'ils permettent et qu'ils offrent la manière économiquement la plus efficace de coordonner des investissements en recherche-développement dans les principales entreprises de production de logiciels. Selon lui, le fait que *VisiCalc* n'ait pas été breveté a plutôt freiné l'inventivité des concurrents qui ont plus cherché à copier cette application qu'à l'améliorer en innovant (p. 198). D'autres, comme Alex Hudson³¹ soutiennent que des brevets pris trop tôt auraient empêché le développement des tableurs. De nombreux procès vont émailler la courte histoire des tableurs dans les décennies quatre-vingt et quatre-vingt-dix.

C'est en s'appuyant sur un dépôt de brevet que René Pardo se présente comme un des co-inventeurs du premier tableur électronique en 1969³². Avec Rémy Landau, ils avaient créé LANPAR (*LANguage for Programming Arrays at Random*), application utilisée pour préparer le budget dans plusieurs entreprises de téléphone comme Bell Canada et AT&T. Selon ces deux auteurs, LANPAR contient deux caractéristiques originales qui sont à la base des tableurs actuels, bien qu'absentes des premiers tableurs comme *VisiCalc* : la déclaration de variables avant leur utilisation (*forward referencing*) et l'utilisation d'un arbre de dépendance entre cellules pour le re-calcul des formules (*natural order recalculation*).

La demande de brevet³³, déposée en 1970, a d'abord été rejetée, puis finalement acceptée en 1983³⁴. On trouve un jugement sur ce brevet, quelques années plus tard, dans une action de la *Refac International, Ltd.* et de *Forward Reference Systems, Ltd.*, contre *Lotus Development Corporation*³⁵. Ce cas est résumé et défendu dans une audition³⁶. Ce brevet, *The automatic recalculation patent for spreadsheets*, est par ailleurs considéré comme un exemple caractéristique de mauvais brevet par Brendan Scott qui en fait une critique amusante sur le site de la *League for Programming Freedom*³⁷.

Fournissant un extrait de ce brevet (claim 30) :

“A process of operating a general purpose data processor of known type to enable the data processor to execute formulas in an object program comprising a plurality of formulas, such that the same results will be produced when using the same given data,

³⁰ <http://www.bricklin.com/patenting.htm>

³¹ Alex Hudson, A future without spreadsheets?
http://www.softwarepatents.co.uk/future/without_spreadsheets.html

³² <http://www.renepardo.com>

³³ <http://patft.uspto.gov/netacgi/nph-Parser?patentnumber=4398249>

³⁴ <http://www.renepardo.com/articles/spreadsheet.pdf>

³⁵ <http://www.ll.georgetown.edu/Federal/judicial/fed/opinions/95opinions/95-1350.html>

³⁶ <http://www.uspto.gov/go/com/hearings/software/arlington/remarks.text> (The Spreadsheet Patent), ACIS Testimony At Pto Hearing On Patent Protection For Software Related Inventions

³⁷ Brendan Scott, What Algorithm Patents Can Do For You, League for Programming Freedom,
<http://lpf.ai.mit.edu/Links/prep.ai.mit.edu/unnatural-order>

regardless of the sequence in which said formulas are presented in said object program, comprising the steps of:

(a) examining each of said formulas in a storage area of the data processor to determine which formulas can be designated as defined;

(b) storing, in the sequence in which each formula is designated as defined, said formulas which are designated as defined;

(c) repeating steps (a) and (b) for at least undefined formulas as many times as required until all said formulas have been designated as defined and have been stored;

whereby to produce the same results upon sequential execution of the formulas stored by said process when using the same given data, regardless of the order in which said formulas were presented in the object program prior to said process.”

Le commentateur de s'extasier ironiquement devant ce qu'il nomme la subtilité du mécanisme de cet algorithme :

“How they ever hit upon the idea of doing first things first and second things second is completely beyond my feeble intellectual capacities.”

Débat entre « précurseurs », construction d'un récit sur Wikipedia

Il est facile d'imaginer qu'un certain nombre d'applications ont pu correspondre à la notion de tableur. La Wikipedia en anglais fournit des éléments intéressants, mais, nous allons le voir, difficiles à attester. En effet, une manière classique d'utiliser les articles de Wikipedia est de les prendre comme pivot, c'est-à-dire d'explorer les liens externes éventuellement proposés et de lancer des recherches à partir des informations données. C'est cohérent avec le fonctionnement déclaré de l'encyclopédie, où tout ce qui est affirmé doit être basé sur des sources publiées. En l'absence de liens externes proposés dans un article, on est contraint à lancer une recherche sur internet.

Dans le cas du tableur (*spreadsheet*), suivre la procédure qui vient d'être décrite, conduit à différents résultats intéressants. Tout d'abord, on s'aperçoit alors que Wikipedia est énormément copiée. Ainsi, lancer des requêtes sur un moteur de recherche avec certains mots spécifiés ne conduit qu'à un ensemble de pages directement reprises de l'encyclopédie. L'annexe I fournit deux exemples avec les requêtes “APL DOT spreadsheet” et “CSSTAB spreadsheet”, deux systèmes présentés comme des précurseurs. Cela conduit à s'interroger sur la difficulté de trouver en ligne des sources originales, quand Wikipedia n'en mentionne aucune. D'ailleurs, l'article mentionne la nécessité de trouver de telles sources.

Ensuite, la rétribution symbolique de pouvoir être considéré comme un des acteurs majeurs de l'histoire du tableur avant *VisiCalc* peut en amener certains à se battre pour faire reconnaître leur droit, dont la légitimité est en question.

Durant l'été 2007, une controverse a émaillé la rédaction de l'article *Spreadsheet*, illustrant d'une part la difficulté, sans étude historique poussée, de reconstituer l'histoire du tableur et d'autre part le fait que Wikipedia peut constituer une arène pour l'écriture de cette histoire, malgré ses principes fondateurs.

La controverse porte sur le rôle qui a pu être joué par *The Works Records Systems (WRS)* une application de la société ICI (Imperial Chemical Industries) dont le concepteur serait Robert Mais et qui aurait été implémentée par une équipe comprenant un certain Ken Dakin (qui semble être à l'origine des informations). Les développements sur *WRS* ont été supprimés de la version du 29 juillet de Wikipedia.

On trouvera en annexe II un résumé de cette diatribe, qui inclut également un autre système nommé *Autoplan/Autotab*. Notons que, bien que ne semblant pas désireux de le faire, Dan

Bricklin, après avoir mis un commentaire sur son blog, a dû intervenir dans la page de discussion de l'article *Spreadsheet* de Wikipedia, montrant certains effets pervers de la construction de cette encyclopédie participative : il est parfois délicat de ne pas y intervenir pour défendre son point de vue.

Ces interventions personnelles d'acteurs directement impliqués dans la conception de tableurs ou de produits apparentés transforme ainsi Wikipedia en un lieu où s'écrit une partie de l'histoire récente, ce qui en fait bien autre chose qu'une « simple » encyclopédie.

D'autres précurseurs de VisiCalc

D'autres produits, tout aussi proches du tableur, pourraient être cités. Ainsi, *IFPS* (Interactive Financial Planning System)³⁸ est développé à la fin des années 1970 par Gerald Wagner et ses étudiants à l'Université du Texas, et mis sur leur marché par la société Execucom Systems. Il répond à l'idée de créer un langage permettant aux cadres financiers de construire des modèles sans l'intermédiaire d'informaticiens professionnels. Selon Power³⁹, l'avantage majeur des langages de planification comme *IFPS* par rapport aux tableurs tient à ce que le modèle est écrit en langage naturel et que ce modèle peut être séparé des données. Plane (1983) constate que l'utilisation répandue des tableurs a rendu le test d'hypothèses (*what if analysis*) presque banal. Mais il argumente en faveur de *IFPS*, ayant plus de possibilités que les tableurs, et déclare qu'il est sans doute utilisé dans plus de 130 universités.

Une investigation plus pointue permettrait d'y voir un peu plus clair, mais on peut se poser la question de l'incidence de tous ces produits sur la « *killer application* » qu'a été *VisiCalc* ? Certains de ces produits étaient connus des concepteurs de *VisiCalc*, plus ou moins imprégnés des outils financiers fonctionnant sur *mainframes*. Mais selon Frankston⁴⁰, le travail de Mattessich n'a pas eu la moindre influence sur *VisiCalc*, puisque ce dernier n'est pas un logiciel comptable mais seulement une application qui permet de faire de la comptabilité.

VisiCalc est-il pour autant le premier tableur ? Dan Bricklin⁴¹ précise qu'il ne s'agit certainement pas du premier programme traitant de tableaux (lignes/colonnes) et fournit un certain nombre de spécificités : un produit interactif de manière WYSIWYG⁴² qui privilégie une « programmation par l'exemple », une interface utilisateur particulière, le fait de tourner sur une machine personnelle peu chère. Comme on l'a vu, *VisiCalc* fut un catalyseur de l'industrie de l'ordinateur personnel, en aidant à l'introduire dans les communautés des affaires et de la finance. Bricklin note également que l'interface utilisateur des tableurs qui ont suivi a conservé une grande compatibilité avec *VisiCalc* comme par exemple, la notation A1 au lieu de R1C1 pour les adresses de cellules, la distinction entre valeurs et formules à la saisie du premier caractère tapé (=), la copie avec adressage absolu et relatif.

Pour Mitch Kapor, c'est bien l'association entre un modèle de données (matrice de cellules avec des formules), qui n'était certainement pas nouveau, et une métaphore de manipulation directe qui a créé cette application majeure⁴³.

³⁸ *What is/was IFPS?* <http://dssresources.com/faq/index.php?action=artikel&id=58>

Interview de Jerry Wagner (12/2003), <http://www.dssresources.com/interviews/wagner/wagner01302004.html>
³⁹ <http://dssresources.com/newsletters/19.php>

⁴⁰ <http://dssresources.com/history/frankston4151999a.html>

⁴¹ <http://www.bricklin.com/firstspreadsheetquestion.htm>

⁴² *What You See Is What You Get*, acronyme forgé à la fin des années 70 pour caractériser une propriété des interfaces

⁴³ <http://dssresources.com/history/kapor4151999.html>

III. De Visicalc à Excel

Selon Campbell-Kelly, (2003, p. 224), la période 1979-1983, pendant laquelle s'est formée l'industrie du logiciel pour micro-ordinateur, ressemble à la ruée vers l'or.

Dès 1980, il y avait environ 75 tableurs en compétition sur le marché, avec des caractéristiques assez semblables. En fait, chacun de ces logiciels devait être conçu en fonction du segment de marché visé, suivant le type de machine et le type de système d'exploitation utilisés. Se posaient ensuite aux concepteurs les questions classiques en matière de fonctionnalités, d'interface utilisateur et de rapidité. Nous ne rappellerons ci-dessous que les plus caractéristiques.

Un article de Myron Berger dans *Personal Computing* d'avril 1982⁴⁴ tente d'expliquer le succès du tableur par sa capacité à répondre aux questions de type « *que se passerait-il si... ?* » (*what-if questions*) c'est-à-dire la possibilité de construire un modèle mathématique, d'identifier les options et les variables et d'analyser l'impact de ces variables sur le modèle. Cet article dresse une liste des vingt principaux tableurs de l'époque (voir annexe III).

Un des plus sérieux concurrents de *VisiCalc* a été *SuperCalc*⁴⁵, produit par la société Sorcim (dont le nom est simplement le mot « micros » à l'envers) pour les ordinateurs tournant sous CP/M. (*op. cit.*, p. 237). En particulier, il était offert avec l'Osborne I, l'ordinateur CP/M le plus populaire à l'époque⁴⁶. Il semble que *SuperCalc* ait été le premier tableur capable de gérer les références circulaires, offrant ainsi une méthode de résolution itérative d'un ensemble de formules.

Multiplan lancé en 1982 par Microsoft pour les PC sous MS-DOS fut le premier tableur apte à gérer plusieurs feuilles de calcul et leurs relations au sein d'un même document.

MBA de Context Management Systems fut le premier logiciel intégré⁴⁷ qui proposait à la fois de la modélisation financière, des graphiques, une base de données et un traitement de textes. Une cellule pouvait contenir jusqu'à quatre pages de texte. Mais sa lenteur lui a été reprochée ainsi que ses limitations dans le traitement du texte⁴⁸.

Mais le principal successeur de *VisiCalc* fut incontestablement *Lotus 1-2-3*, dont les premiers exemplaires sont arrivés sur le marché en janvier 1983.

Lotus 1-2-3 et ses concurrents

Si *VisiCalc* a fait une partie du succès de l'Apple II, *Lotus 1-2-3* a tout autant contribué à celui des PC dont il a largement dominé le marché pendant plusieurs années. Il se présentait comme un intégré, regroupant un tableur, un gestionnaire de données et un générateur de graphiques. Lotus Development Corporation⁴⁹ a été fondée par Mitch Kapor qui a conçu *Lotus 1-2-3*, implémenté pour IBM PC par Jonathan Sachs (ainsi que George Riner et Rick Ross). On peut suivre le passage de *VisiCalc* à *Lotus* dans un entretien avec Mitch Kapor pour le Computer History Museum⁵⁰.

⁴⁴ <http://www.aresluna.org/attached/computerhistory/articles/spreadsheets/scenariosforsuccess> : *Scenarios for Success: The Vision Of Spreadsheets*.

⁴⁵ <http://en.wikipedia.org/wiki/SuperCalc>

⁴⁶ http://www.hodgy.net/computer_history/page_3/page_3.htm

⁴⁷ http://www.dvorak.org/blog/?page_id=8250 *Whatever Happened to Context MBA?* John C. Dvorak

⁴⁸ <http://www.secretguide.net/read/index.php?filename=spreadsheets>

⁴⁹ Le nom choisi par Mitch Kapor vient du fait qu'il a donné des cours de méditation transcendantale et qu'il entrainait en extase dans la contemplation des fleurs de lotus.

⁵⁰ http://www.softwarehistory.org/history/mitch_kapor.htm

Initialement, le programme *1-2-3* était censé gérer des feuilles de calcul, des graphiques et un traitement de texte. Mais Jonathan Sachs, instruit par l'exemple de *MBA* de Context, comprit qu'intégrer un bon traitement de texte serait trop consommateur en mémoire de travail et ralentirait trop le programme. Aussi remplaça-t-il le traitement de texte par un gestionnaire de bases de données⁵¹. *Lotus 1-2-3* est arrivé sur le marché en janvier 1983 et a rapidement rencontré un grand succès. Quelles sont les caractéristiques principales de ce nouveau produit⁵² ? Outre les nouvelles fonctions liées au graphique et à la gestion de bases de données, Lotus apparaît spécialement conçu pour être facilement utilisé, intégrant un tutoriel interactif et des fichiers d'aide, le rendant à la fois puissant et facile à utiliser, même pour des débutants. On pourra noter que *Lotus* est une des premières applications à fournir une aide contextuelle à l'utilisateur, accessible par une touche de fonction (F1), dispositif maintenant courant dans les applications de bureautique, et du point de vue du calcul, qu'il met en œuvre un arbre de dépendance entre cellules pour le re-calcul des formules.

En 1985, Lotus Development Corporation achète la société Software Arts, la compagnie fondée par Bricklin et Frankstone, et arrête la diffusion de *VisiCalc*.

Portés par le succès de Lotus 1-2-3, plusieurs éditeurs ont commercialisé des modules complémentaires qui enrichissent et spécialisent le tableur d'origine. Parmi ceux-ci, il faut mentionner Frontline Systems⁵³, entreprise dirigée par Dan Flystra qui a développé en 1990 *What-If Solver*, fonctionnalité permettant de résoudre des problèmes linéaires ou non linéaires, avec contraintes sur les variables. Il s'agit sans doute d'un des premiers *solveurs* associés à un tableur, toujours présent dans plusieurs produits actuels, Excel 2007 compris⁵⁴.

À la suite de Lotus, de nombreux tableurs vont être proposés, ainsi que des logiciels intégrés, c'est-à-dire offrant comme *Context MBA* et *Lotus 1-2-3* des fonctionnalités de traitement de texte, de gestionnaire de bases de données... Un article de Rich Malloy, publié à l'été 1987 dans *Byte*, présente 9 tableurs, dont *Excel* que nous verrons plus loin, et compare leurs principales caractéristiques (voir Annexe IV).

Parmi les concurrents de Lotus, il faut citer *Quattro* qui, selon John Walkenbach, a toujours été le troisième tableur⁵⁵. Ce produit, dont la première version est sortie en 1987, a originellement été développé par Borland International (puis vendu en 1994 à Novell et peu après à Corel). *Quattro* correspond au chiffre 4, pour montrer la succession de *Lotus 1-2-3*. D'ailleurs, selon une blague rapportée dans la foire aux questions sur les tableurs⁵⁶, le nom de code pour *Quattro* était Buddha, du fait qu'il devrait prendre la position du Lotus !

Cette guerre commerciale, par ailleurs bien documentée, n'est pas notre propos mais révèle aussi l'impact des succès commerciaux sur les fonctionnalités du tableur qui, progressivement vont orienter l'évolution de cet objet qu'est le tableur et ses futurs possibles.

⁵¹ <http://www.secretguide.net/read/index.php?filename=spreadsheets>

⁵² Gregg Williams, A fast, versatile package that combines spreadsheet, graphing, and database functions: Lotus Development Corporation's 1-2-3, *Byte*, décembre 1982, <http://www.aresluna.org/attached/computerhistory/articles/spreadsheets/lotus123review>

⁵³ <http://www.solver.com/presshist.htm>

⁵⁴ La présentation du solveur associé à Excel sur le site de Microsoft fait explicitement référence à Frontline Systems : <http://office.microsoft.com/fr-fr/excel/HP100726911036.aspx>

⁵⁵ <http://j-walk.com/ss/qpro/index.htm>

⁵⁶ <http://www.faqs.org/faqs/spreadsheets/faq/>

Lotus Improv

Bien qu'il n'ait pas rencontré de succès au plan commercial, *Improv* semble occuper une place particulière. Selon Christopher Browne⁵⁷, il s'agissait du meilleur modèle de gestion et de construction de tableur. Bill Jelen, un expert reconnu du tableur *Excel*, et auteur du site MrExcel.com est du même avis (communication personnelle).

On trouve différentes sources sur l'histoire d'*Improv*⁵⁸. Il s'agissait de séparer les données, les vues sur les données et les formules et de réaliser un produit s'inscrivant véritablement dans le paradigme objet⁵⁹. C'est pourquoi *Improv* a été conçu pour la machine NeXT, dont le système d'exploitation repose sur des objets.

Dans un tableur classique, une formule est associée à une cellule et pour l'appliquer à plusieurs cellules, il est nécessaire de la recopier autant de fois que nécessaire. Dans *Improv*, les formules ne sont pas associées aux cellules mais sont des objets de base associés à la feuille de calcul ; elles s'appliquent à un ensemble de cellules et peuvent opérer sur des items, des groupes et des catégories. Les formules utilisent des noms à la place des adresses de cellules, ce qui les rend plus compréhensibles, surtout pour les utilisateurs qui ne sont pas les concepteurs des feuilles de calcul.

L'autre caractéristique importante d'*Improv* est de séparer les données des vues sur les données. En effet, avec *Improv* apparaissent les tableaux croisés dynamiques (*pivot tables*) qui sont une des fonctionnalités avancées importantes d'*Excel*. L'idée est de réorganiser et résumer des sélections de lignes et de colonnes de données pour obtenir un rapport centré sur des éléments significatifs du tableau. Il s'agit d'obtenir une vue particulière, sans changer les données de la feuille de calcul. L'approche objet d'*Improv* lui permettait de réaliser facilement ce type de transformation.

Improv a été disponible en février 1991 pour la machine NeXT. La version Windows est sortie en 2003, mais elle a rencontré des résistances importantes.

En 2004, Dam Bosworth lors de la conférence ICSOC'04, explique cet échec par le fait de la réussite même du projet d'*Improv*⁶⁰.

“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.”

Le concepteur des tableaux croisés dynamiques d'*Improv*, Pito Salas, répondant à l'auteur de cette explication⁶¹, tente de trouver des raisons à cet échec. En effet, il reconnaît que l'une des caractéristiques du tableur est sa « *malléabilité* », et que celle-ci permet, sous l'action de l'utilisateur, qu'une feuille de calcul puisse ainsi se développer presque « *organiquement* », de façon totalement « *improvisée* », jusqu'à ce qu'elle en devienne parfois un vrai « *fouillis* ». Ce qu'*Improv* a tenté d'éviter et en quoi il a échoué, toujours selon Pito Salas. On pourrait ainsi dire que la tentative que représente *Improv* d'améliorer le cadre formel de conception

⁵⁷ <http://cbbrowne.com/info/spreadsheets.html>

⁵⁸ Notamment celui de Simson Garfinkel, informaticien, auteur de nombreux ouvrages et chroniqueur, <http://www.simson.net/clips/91.NW.Improv.html>

⁵⁹ On peut citer Piersol (1986) qui décrit un tableur orienté objet en *Smalltalk*.

⁶⁰ <http://www.adambosworth.net/archives/000031.html>

⁶¹ <http://www.salas.com/weblogs/archives/000522.html> et <http://www.salas.com/weblogs/archives/000524.html>

d'une feuille de calcul aura ainsi montré quelles contraintes la majorité des utilisateurs est prête à accepter et celles qu'elle refuse, pour ne pas perdre de « malléabilité ».

Excel

Le produit qui domine aujourd'hui incontestablement le marché, et ce depuis le début des années 90 est un tableur « classique », *Microsoft Excel*. Comment expliquer cette aussi longue période de succès ?

On pourra se reporter à deux études qui font référence, la première de Campbell-Kelly (2001) sur le développement de l'industrie du logiciel pour PC, et la seconde de Liebowitz et Margolis (1999) qui analysent l'évolution du marché des tableurs dans les années 80 et 90 selon un point de vue à la fois technique et économique .

Commençons par souligner la faiblesse des sources historiques fournies directement par Microsoft. Nous avons déjà mentionné la disparition de ressources en ligne regrettée par John Walkenbach⁶². L'hypothèse que l'on peut formuler – en remerciant Bill Jelen de nous l'avoir suggérée – est que Microsoft redoute, en prétendant avoir inventé telle ou telle chose, que des « inventeurs », pouvant attester d'une certaine antériorité, fassent des recours devant les tribunaux pour en tirer des bénéfices.

Reprenons d'abord la suite des événements pour voir comment Excel a peu à peu conquis le marché.

En 1983, année du lancement de Lotus 1-2-3, Apple crée *LisaCalc*⁶³, le premier tableur utilisant une souris, tournant sur Lisa, un des premiers micro-ordinateurs dont le système d'exploitation adopte d'emblée une interface graphique. Au lancement du MacIntosh, successeur de Lisa en 1984, Microsoft propose d'abord une version de *Multiplan* puis dès l'année suivante, un nouveau tableur nommé *Excel* (parce qu'excellent) qui, comme *Lotus 1-2-3* gère des feuilles de calcul, des graphiques et des bases de données. Un peu plus tard en 1987 (car il semble qu'un accord a été passé entre Apple et Microsoft pour interdire un portage trop rapide d'*Excel* sur Windows), Excel est également proposé pour le PC⁶⁴.

Il est important de noter qu'*Excel* a été conçu d'emblée pour un système d'exploitation graphique. Selon Bill Jelen, *Excel* l'emporte sur ses concurrents, *Lotus* et *Quattro*, en termes de fonctionnalités dont beaucoup restent probablement encore inconnues des utilisateurs actuels, mais apparaît beaucoup plus lent à cause des machines qui ne sont pas encore suffisamment performantes. Ainsi, un article de Jared Taylor paru en décembre 1987 dans *PC Magazine*⁶⁵, décrit *Excel* comme bien plus puissant que *Lotus 1-2-3*, mais souligne le fait que nombre d'utilisateurs ne sont pas prêts à payer le prix élevé d'une machine assez puissante pour le faire tourner. On trouve la même remarque dans un autre article de Rich Malloy dans *Byte* : *Microsoft's Macintosh spreadsheet shines on the IBM AT, but is it fast enough?*⁶⁶.

En pratique, le débat porte sur le temps de recalcul des cellules. *Excel* pouvait apparaître plus rapide qu'il ne l'était en réalité, du fait qu'il rendait la main à l'utilisateur qui pouvait ainsi continuer à saisir des valeurs avant que le recalcul ne soit terminé. Bill Jelen (communication personnelle, 23 août 2007) souligne que cette fonctionnalité était très dangereuse, puisqu'il

⁶² <http://www.j-walk.com/ss/history/index.htm>

⁶³ Présentation commerciale de LisaCalc,
<http://www.guidebookgallery.org/ads/magazines/lisa/lisacalc>

⁶⁴ <http://www.secretguide.net/read/index.php?filename=spreadsheets>

⁶⁵ <http://www.aresluna.org/attached/computerhistory/articles/spreadsheets/anewfaceforspreadsheets>

⁶⁶ <http://www.aresluna.org/attached/computerhistory/articles/spreadsheets/excelexttraordinaire>

était possible d'interrompre ce recalcul par mégarde et de se tromper sur la valeur des cellules qui n'auraient pas été actualisées.

Mais cela n'empêche pas Microsoft de conserver sa stratégie à moyen terme et de continuer d'ajouter des fonctionnalités, qui deviendront efficaces avec l'évolution de la puissance des ordinateurs.

La version 3 de *Lotus 1.2.3* était un programme pour DOS tournant sur Windows, et la version 4 fut la première version écrite réellement pour Windows. Elle sort en 1993, huit ans après le premier tableur graphique de Microsoft. Dans la lutte pour la conquête des parts de marché entre 1993 et 1997, les grandes caractéristiques des tableurs sont intégrées à Excel et à partir de 1997, la cause est entendue. On trouvera sur le site de Walkenbach, la liste des dix versions successives d'Excel⁶⁷ qui sont sorties sur le marché entre 1985 et 2007.

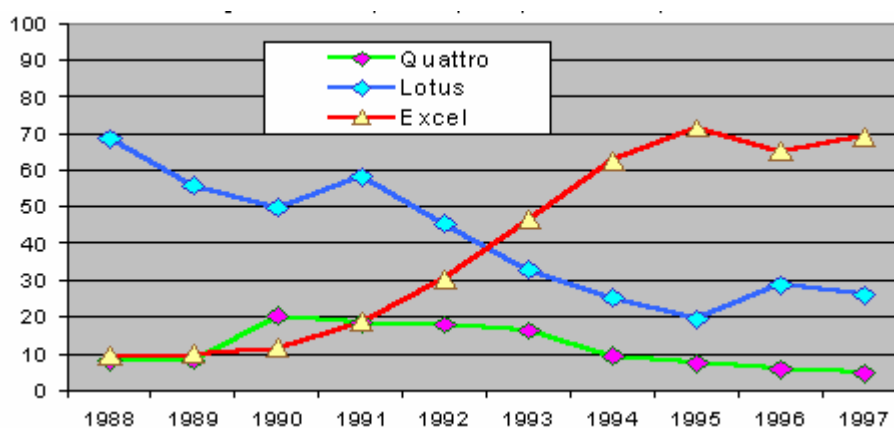


Figure 4. Partage du marché des tableurs en unités vendues (extrait de Liebowitz et Margolis, 1999).

Pourquoi une telle domination ? Ce que les économistes appellent *l'effet réseau*, qui conduit chaque utilisateur à choisir de préférence ce que les autres ont déjà, a probablement joué un rôle déterminant. Le fait d'écrire des formules et des macros avec une syntaxe précise et spécifique, ce qui alourdit la charge de la migration d'un tableur à un autre, explique aussi pourquoi nombre d'utilisateurs restent longtemps fidèles à une marque de produits compatibles entre eux (Liebowitz et Margolis, 1999).

Une deuxième explication tient dans la lutte entre les machines et les systèmes d'exploitation (Campbell-Kelly, 2001). C'est le succès de Windows et du PC qui a permis aux logiciels de Microsoft de dominer le marché alors que ses concurrents avaient parié sur d'autres machines (NeXT choisi par Lotus pour Improv) ou d'autres systèmes d'exploitation (comme OS/2 d'IBM). C'est bien la capacité de Bill Gates à anticiper ce qui va fonctionner, mais aussi son opiniâtreté à relancer plusieurs versions successives d'un même produit, qui lui permet d'emporter le marché. La vitesse de calcul étant très importante pour les tableurs (des benchmarks y étaient consacrés pour comparer les différents produits, importance du temps de recalcul avec le *what if?*), le marché s'est longtemps réparti entre les machines et les systèmes d'exploitation (ce qui a fait un temps le succès de SuperCalc pour CP/M). Dès que les machines ont été assez puissantes pour supporter les interfaces graphiques, Microsoft avait acquis une avance importante sur ces concurrents. Mais ce n'est que la version 5 d'Excel, en 1993, qui est devenue majoritaire sur le marché.

Une autre raison peut être trouvée dans l'extension progressive du public visé et finalement atteint, composé d'utilisateurs plus néophytes, moins formés, intéressés par les produits les

⁶⁷ <http://j-walk.com/ss/excel/index.htm>

plus simples d'emploi. Michael Antonoff note dans un article dans *Personal Computing* (novembre 1986), que la manière dont on considère les nombres a changé avec les tableurs⁶⁸, et que les catégories de personnes utilisant les tableurs dans les entreprises ont changé, passant des comptables et analystes financiers aux secrétaires et employés de bureau (office administrators), cette seconde vague ayant la charge de saisir les données pour leurs responsables.

Même si des programmes tels *Lotus Improv* pouvaient disposer de fonctionnalités à l'époque très en avance, c'est bien *Excel* qui a peu à peu a conquis le marché et progressivement inclus des fonctionnalités très avancées dans un produit plus « grand public ». Ce qui en ferait un des grands logiciels jamais écrits d'après Charles Babcock, chroniqueur à *Information Week*⁶⁹. Mais que sait-on des utilisations des tableurs ?

IV. Utilisations et questions actuelles

Le tableur a peut-être révolutionné l'histoire du calcul mais il a aussi modifié la manière d'aborder le traitement des nombres, comme l'exprime Alain Le Diberder dans sa définition du mot tableur⁷⁰ :

« On considère souvent que le tableur est un "traitement de chiffres", au même titre qu'il y a des traitements de textes. Mais le tableur fait subir aux chiffres bien plus que ce que le traitement de texte fait aux mots. Il les travaille, les ordonne, en fait naître d'autres, et leur donne un nouveau sens. »

Dans un cours destiné à des enseignants de lycée intitulé *How Computers Have Simplified Accounting*⁷¹, Carol L. Cook du Yale-New Haven Teachers Institute, souligne que le tableur est plus rapide, plus exact et plus facile que les techniques comptables traditionnelles et qu'il est devenu le logiciel le plus important dans le monde des affaires. Ses utilisations sont variées et toujours d'actualité.

Ainsi, dans un article du *Journal du Net* publié en 2006⁷², Dominique Filippone considère que le tableur reste un outil indispensable pour l'analyse décisionnelle. Une enquête menée par *Compass*⁷³ auprès d'une trentaine de grandes entreprises indique que le tableur (*Excel* en l'occurrence) est toujours utilisé par la plupart d'entre elles pour préparer leurs tableaux de bord.

La création de tableaux dynamiques répond bien aux exigences des directions financières et marketing ; le tableur est un outil ordinaire pour l'établissement de rapports (reporting). Mais des limitations sont mises en exergue : faible maîtrise de son évolutivité (qui aboutit à des « usines à gaz »), capacité limitée d'interopérabilité (comment faire le lien avec les modèles web), faible degré de partage d'informations. Limitations que certains éditorialistes considèrent comme un cauchemar : le *Spreadsheet Hell*⁷⁴.

Mais qui sont donc les utilisateurs ordinaires des tableurs ?

⁶⁸ <http://www.aresluna.org/attached/computerhistory/articles/spreadsheets/thewaywelookatthenumbers>

⁶⁹ What's The Greatest Software Ever Written? *Information Week*, août 14, 2006 12:01 AM
<http://www.techweb.com/article/showArticle.jhtml?articleId=191901844&pgno=1>

⁷⁰ Alain Le Diberder, *Sabir Cyber* <http://www.clve.fr/sabircyber/tableur.htm>

⁷¹ <http://www.yale.edu/ynhti/curriculum/units/1989/7/89.07.06.x.html>

⁷² http://www.journaldunet.com/solutions/0602/060203_analyse-tableur-decisionnel.shtml

⁷³ Étude citée par ZDNET <http://www.zdnet.fr/actualites/it-management/0,3800005311,39371121,00.htm> et disponible à http://www.compassmc.com/pdf/Dashboard_evaluation_deliverables.pdf

⁷⁴ Don Durfee sur CFO.com : <http://www.cfo.com/article.cfm/3014451>

Des utilisateurs formés sur le tas

Différentes études ont traité des utilisations professionnelles des tableurs. La plupart se sont limitées à de petites populations d'utilisateurs. Ainsi, dans l'introduction de sa thèse qui porte sur la visualisation des structures régulières dans les tableurs, Markus Clermont (2003, p. 13) distingue quatre catégories de feuilles de calcul qu'il associe aux statistiques, au calcul scientifique, à la gestion et à l'algorithmique. Les utilisateurs en sont, respectivement, des scientifiques, des ingénieurs, des gestionnaires et des informaticiens. Dans un tout autre cadre, Grenville Croll (2005) donne un aperçu des utilisations dans le monde des grandes organisations qui opèrent au cœur de la City à Londres, en montrant que les tableurs jouent un rôle essentiel dans certains secteurs financiers et que les praticiens confirmés ne contrôlent pas toujours les risques encourus par les erreurs qui peuvent y être introduites.

À notre connaissance, une des études les plus complètes sur les usagers du tableur a été conduite par le *Spreadsheet Engineering Research Project*⁷⁵ (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 entreprises. Un questionnaire en ligne comprenant 67 items a été rempli en 2005 et 2006 par près de 1600 répondants appartenant à des institutions différentes du monde des affaires et de l'informatique ; ces résultats fournissent une image très riche des pratiques des concepteurs et utilisateurs de feuilles de calcul⁷⁶.

Ce travail confirme d'abord l'écrasante domination d'Excel, puisque la liste des logiciels utilisés fait apparaître la hiérarchie suivante :

Microsoft Excel	1586	99.3%
Lotus 1-2-3	39	2.4%
Quattro Pro	24	1.5%

Tableau 1 : Types of software you use in your job

Plus de 80% des répondants considèrent que les tableurs ont un rôle très important, voire critique 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 utilisateurs ont appris à se servir du tableur par différentes voies. Moins de 40% 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 ont joué un rôle pour plus de la moitié des utilisateurs. Comme l'avaient remarqué Nardi et Miller (1990a) dans une étude ethnographique, il peut y avoir des formes de collaboration de personnes ayant des niveaux d'expertise en programmation très différents.

None	281	17.6%
Formal classroom instruction	602	37.7%
Occasional informal training sessions	467	29.2%
Books and manuals	856	53.6%
Demonstrations from colleagues	835	52.3%

Tableau 2 : Type(s) of training have you had using spreadsheets

⁷⁵ http://mba.tuck.dartmouth.edu/spreadsheet/full_project_description.html

⁷⁶ Les publications et les résultats de cette enquête sont disponibles à http://mba.tuck.dartmouth.edu/spreadsheet/product_pubs.html

Toutefois, ces personnes travaillent seules pour la très grande majorité et plus de la moitié consacre au tableur plus du quart de son temps de travail.

By yourself	1289	81.1%
In a team of 2 or 3	259	16.3%
In a larger team (4 or more)	42	2.6%

Tableau 3 : When working with spreadsheets, you typically work

0-25%	712	44.7%
26-50%	484	30.4%
51-75%	284	17.8%
76-100%	114	7.2%

Tableau 4 : Approximate percent of time spent with spreadsheets in your job

***End user programming* ou quels programmeurs ?**

Le rapport du tableur à la programmation est ambigu. D'une part, le recours au tableur permet d'éviter de se lancer dans des développements complexes. C'est explicite dès l'origine, notamment au milieu des années 80, par exemple pour la programmation de simulations (Reese, 1984). Et cela semble toujours vrai, comme le rappelle Thiriez (2004) à propos de la modélisation professionnelle, ou d'autres simulations comme celles des files d'attente (Leong, 2007). Ainsi, selon Thiriez, les modèles Excel sont très intéressants pour leur interactivité, leur temps de développement et leurs coûts réduits, leur convivialité, l'attitude positive des clients et des utilisateurs (le modèle n'est pas une boîte noire), les riches possibilités graphiques (quand on ne se limite pas à ce qu'offre Excel en standard).

Cette possibilité de programmer, largement employée par des utilisateurs qui ne sont pas des informaticiens et qui ne souhaitent pas le devenir, a suscité un courant d'études et de recherche nommé « *end user programming* »⁷⁷, terme qui n'a pas donné lieu à équivalent précis en français. On retrouve le tableur, mais surtout *VisiCalc* aux origines de ce courant, comme le rappellent Peter Hale (University of the West of England) sur son blog⁷⁸ ou Trevor Smedley (Dalhousie University à Halifax) sur son site⁷⁹.

Selon Nardi et Miller (1990b), les systèmes de programmation par l'utilisateur final (*end user programming systems*) permettent à ces utilisateurs de créer des applications utiles avec une formation ne dépassant pas une poignée d'heures. L'interface utilisateur y joue un rôle déterminant, ce qui conduit les chercheurs à s'intéresser aux langages de programmation visuels. Mais rendre les choses visuelles ne suffit pas toujours à rendre un système réellement utilisable, problème qui reste posé dans un champ de recherche en plein essor.

On peut s'interroger sur l'ampleur de cette activité de programmation. Ainsi, Scaffidi *et al.* (2005), reprenant un travail de Boehm publié en 1995, essayent d'évaluer le nombre de *end users programmers*. Ils obtiennent, pour les États-unis, en 2012, une estimation de 90 millions d'utilisateurs d'ordinateurs sur le lieu de travail, dont 55 millions utilisant des tableurs et des gestionnaires de bases de données et environ 13 millions de « programmeurs ».

⁷⁷ Voir par exemple les actes du Workshop on End-user software engineering - CHI 2006 : <http://eusesconsortium.org/weuse/proceedings.php>

⁷⁸ <http://userdrivenmodelling.blogspot.com/2007/05/history-of-end-user-programming.html>

⁷⁹ <http://users.cs.dal.ca/~smedley/Research.html>

Apprentis sorciers ?

On a fourni une sorte d'ardoise magique pour les calculs. Mais n'a-t-on pas transformé des utilisateurs en apprentis sorciers, donnant des outils professionnels à des non-professionnels, ou plutôt à des non-programmeurs ? Cela ne va-t-il pas provoquer des désastres ? La sonnette d'alarme a déjà retenti : d'une part pour la programmation utilisateur (voir par exemple Harrisson, 2004), mais depuis longtemps pour les tableurs.

Ce côté programmation offert à des non-professionnels recèle des risques qui ont été analysés assez tôt. 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 qui a longtemps analysé ce phénomène⁸⁰ 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)⁸¹ qui organise une conférence annuelle et dont le site contient une page spéciale sur les erreurs contenues dans des tableurs dont les conséquences sont plus remarquables⁸².

Dans deux comptes rendus de la conférence EuSpRIG de 2007 par Patrick O'Beirne⁸³ et John Stokdyk⁸⁴, 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.

Des méthodes plus rigoureuses provenant du génie logiciel peuvent-elles être adoptées par des utilisateurs qui ne sont pas des informaticiens ? C'est un des thèmes de travail du collectif *End Users Shaping Effective Software* (EUSES)⁸⁵, et aussi un sujet débattu dans un atelier de CHI 2006⁸⁶ ou encore dans une thèse récente sur *End-user software engineering in the spreadsheet paradigm* (Robin, 2007).

Proposer des environnements de programmation ou des fonctionnalités de tableurs facilitant le travail de programmation et son contrôle, est une orientation suggérée, avec une attention particulière portée sur la visualisation. Ainsi, Ballinger *et al.* (2003) explorent différentes techniques pour représenter le contenu d'une feuille de calcul ; Igarashi *et al.* (1998) proposent des visualisations statiques et dynamiques pour diminuer la charge cognitive des utilisateurs ; Boulicaut *et al.* (1998) proposent de générer les feuilles de calcul à partir de l'énoncé de spécifications sous la forme de règles ; Clermont (2003) tente de fournir un format plus proche du domaine traité ; Burnett *et al.* développent avec le langage Forms/3⁸⁷ des méthodologies de test visuel pour aider les utilisateurs à vérifier l'exactitude de leurs feuilles de calcul.

⁸⁰ Robert Panko (2000). What We Know About Spreadsheet Errors
<http://panko.shidler.hawaii.edu/SSR/Mypapers/WhatKnow.htm>

⁸¹ <http://www.eusprig.org/>

⁸² <http://www.eusprig.org/stories.htm>

⁸³ <http://www.eusprig.org/conf2007report.htm>

⁸⁴ Spreadsheet error research: Wasted time worse than mistakes. By John Stokdyk, 26 juillet 2007
<http://www.accountingweb.co.uk/cgi-bin/item.cgi?id=171091&d=526&h=524&f=525>

⁸⁵ <http://eusesconsortium.org/>

⁸⁶ Workshop Report: From End-User Programming to End-User Software Engineering (a CHI'06 Workshop)
<ftp://ftp.cs.orst.edu/pub/burnett/weuseII-TR-cs07-60-04.pdf>

⁸⁷ Présentation du langage Forms/3 et publications associées
<http://web.engr.oregonstate.edu/~burnett/Forms3/forms3.html>

Certains auteurs se posent des questions sur le rôle de la formation initiale et continue des utilisateurs. Ainsi, le collectif EUSES consacre une partie de ses activités à promouvoir un enseignement du tableur en mathématiques au lycée⁸⁸.

Le tableur en éducation

Dans leur revue de 25 années d'utilisation des tableurs en éducation, Baker et Sugden (2003) rappellent que, dès 1984, des enseignants discutaient déjà de l'apport du tableur à l'enseignement des mathématiques.

Un des premiers domaines concernés est celui de l'enseignement de statistiques. Pour Sober et Lee (1985), la présentation en lignes et colonnes des données et des valeurs calculées permet une inspection et des modifications aisées pour démontrer différentes propriétés statistiques.

Très tôt apparaît un débat sur l'emploi respectif des tableurs et des langages pour enseigner la programmation et l'informatique (Hsiao, 1985, cité par Baker et Sugden 2003). Shinnars-Kennedy présente en 1986 un projet dans lequel le tableur est employé pour simuler un ordinateur. En France, on s'intéresse aux liens entre tableurs et programmation objet (voir par exemple, Favre-Nicolin, 1988).

Mais c'est avant tout en mathématiques que l'on va trouver de nombreuses références, pour la plupart en langue anglaise, à l'exception de Capponi et Balacheff (1989) : théorie des nombres, combinatoire, analyse numérique, statistiques. Viennent ensuite les sciences expérimentales et toutes les applications pour le traitement et la représentation des mesures et les simulations ; ainsi le quart de leurs références bibliographiques de Baker et Sugden (2003) se rattachent pas à ces utilisations. Bien entendu, le tableur est aussi utilisé pour l'enseignement de l'économie ou de la recherche opérationnelle.

Vers un nouveau paradigme : les tableurs gratuits en ligne

Pour Campbell-Kelly, alors que tant d'exemplaires se sont vendus de par le monde depuis *VisiCalc*, le tableur est devenu un logiciel d'une telle complexité que peu d'utilisateurs peuvent assurer que leurs feuilles de calcul ne contiennent pas d'erreur et que le nombre de ceux qui sont aptes à saisir toutes les potentialités offertes par ces logiciels est encore plus réduit. D'après lui, on manque sans doute d'un modèle et d'une théorie des tableurs. Ainsi, les tableurs modernes ne seraient pas le produit d'un processus cohérent de conception mais plutôt un agrégat de fonctionnalités sélectionnées par des millions d'utilisateurs selon un processus de type darwinien (*The Rise and Rise of the Spreadsheet*⁸⁹).

Des nouveaux modèles sont proposés, reprenant par exemple les idées de Lotus *Improv*, ou fondés sur d'autres paradigmes. Christopher Browne⁹⁰ résume ainsi les points noirs des tableurs traditionnels : ils encouragent une programmation spaghetti, permettent les erreurs, sont difficiles à vérifier, ne fournissent pas de bons outils de gestion de données et, malgré leur commodité, ne constituent pas un dépôt robuste pour l'information. D'après lui, il faudrait faciliter et encourager l'extraction de données à partir de sources externes.

⁸⁸ <http://eusesconsortium.org/edu/education.php>

⁸⁹ Cf. résumé d'un séminaire donné en 2003

<http://www.swan.ac.uk/compsci/research/seminars/seminar.php?seminar=83>

⁹⁰ <http://cbbrowne.com/info/spreadsheets.html>

Jocelyn Paine, dans sa lettre d'information AI Expert de mai 2005⁹¹, décrit différentes extensions intéressantes à la notion de tableur, dont une consiste à ajouter des règles aux feuilles de calcul en utilisant Prolog comme langage de manipulation de modèles.

À partir du milieu des années 90, le déploiement d'Internet contribue à changer la question. Dès 1996, Scott McNealy, le président de Sun, prévoit une évolution vers un *websheet*, tableur léger qui communique via Internet pour récupérer et échanger des données en temps réel⁹².

Dix ans plus tard, on assiste peut-être à une révolution. Alors que le marché de la bureautique a été un moment calqué sur celui des systèmes d'exploitation, celle des applications en ligne est en passe de bouleverser les tendances. Fonctionnant sur un simple navigateur, elles ne nécessitent aucune installation spécifique sur le poste de travail et sont donc indépendantes des plates-formes des utilisateurs.

Cette émergence est liée aux avancées techniques qui permettent de créer des applications web interactives en partageant le traitement entre le serveur et le navigateur. Accessibles depuis n'importe quel poste connecté au réseau mondial, les outils et les services en ligne qui les accompagnent offrent un espace de stockage et des fonctions de partage et de collaboration. On assiste à une effervescence qui pourrait rappeler celle des années 80, à ceci près que les innovations ne portent pas tellement sur l'interface, souvent similaire à celle des outils classiques, ni sur le calcul, ni sur les fonctions, la plupart des outils reprenant les plus classiques et les plus courantes, mais plutôt sur les possibilités de partage et de collaboration. Cette offre abondante intègre souvent plusieurs applications pour constituer des suites bureautiques en ligne. Parmi les plus populaires, on peut citer : *Google Document et Tableur*⁹³, *Zoho*⁹⁴, *ThinkFree*⁹⁵, *NumSum*⁹⁶.

Cette rupture technologique s'accompagne aussi d'une (r)évolution sociale. Ainsi une entreprise comme eXpresso⁹⁷ propose aux utilisateurs d'Excel des espaces collaboratifs offrant garanties de sécurité et facilités d'échange, espace personnel, espaces de collaboration et outils de comparaison et de conversion de feuilles de calcul, sans qu'ils aient besoin d'abandonner leurs solutions de bureau traditionnelles.

Mentionnons au passage *WikiCalc*⁹⁸, une idée originale de Dan Bricklin associée à *Socialcalc*⁹⁹, laquelle consiste à combiner un wiki, utilisé pour sa facilité d'édition par des auteurs multiples, et un tableur, pour sa forme visuelle et ses techniques de calcul familières. L'approche wiki complétée par un accès contrôlé aux documents permet ainsi à plusieurs utilisateurs de créer et de modifier aisément des pages contenant des feuilles de calcul.

L'explosion des services en ligne remet en avant le débat sur l'interopérabilité et les formats d'échange de tableurs. En parallèle à *Open XML*, format propriétaire de Microsoft, se dégage un autre format ouvert, *Open Document*, qui a fait l'objet d'une norme ISO¹⁰⁰ et qui a été adopté par plusieurs éditeurs de suites bureautiques parmi lesquels Google et surtout Open

⁹¹ <http://www.j-paine.org/ainewsletter/may2005.html>

⁹² From Spreadsheet to Websheet, <http://cgi.netscape.com/comprod/columns/intranet/sun1.html>

⁹³ <http://docs.google.com/>

⁹⁴ <http://www.zoho.com/>

⁹⁵ <http://product.thinkfree.com/>

⁹⁶ <http://www.numsum.com/>

⁹⁷ <http://www.expressocorp.com/>

⁹⁸ <http://www.softwaregarden.com/products/wikicalc/>

⁹⁹ <http://www.eu.socialtext.net/open/index.cgi?socialcalc>

¹⁰⁰ ISO/IEC 26300:2006

http://www.iso.org/iso/fr/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43485

Office¹⁰¹, communauté représentative du courant du logiciel libre, rejointe par IBM en septembre 2007¹⁰². Faut-il voir dans l'adoption de ce format une influence des utilisateurs devenus plus attentifs aux possibilités d'échange et aux formats qui les facilitent, les contraintes de l'échange l'emportant alors sur celles de la production ?

V. Perspectives

Le tableur est incontestablement très présent dans nombre d'entreprises et fait partie du quotidien de quantité d'employés et de cadres. Que le dessinateur Scott Adams y ait consacré encore récemment une série de son personnage Dilbert en est un signe révélateur¹⁰³.

Quand on essaye de renouer les fils de l'histoire du tableur, ce n'est pas tant son aspect comptable qui apparaît le plus important que l'outil de calcul fondé sur une organisation spatiale en tableau. Selon Campbell-Kelly (2005), le tableur n'est qu'une solution parmi d'autres au problème d'un dispositif de calcul généralisé, mais sa large acceptation et son ubiquité ont évacué les solutions alternatives. Pour Nardi et Miller (1990b), le tableur offre une représentation concrète et visible des valeurs de données, une réaction immédiate aux actions des utilisateurs ainsi que de puissantes fonctionnalités telles que l'application de formules à des blocs de cellules. Il représente en cela un bon compromis entre simplicité et expressivité.

Deux propriétés essentielles rendent son utilisation par des non-informaticiens bien adaptée à la résolution de problèmes courants : (1) des techniques de calcul relativement simples passant par des fonctions associées directement aux données, évitant ainsi de régler des détails de bas niveau qui sont une limitation de la programmation traditionnelle ; (2) une interface sous forme de tableau qui sert de modèle général à toutes les applications. Ainsi, dans ce type de programmation fonctionnelle, les modes de contrôle du traitement ne sont pas absents puisqu'ils peuvent être appliqués localement. Et la forme *tableau* permet aux utilisateurs de régler trois difficultés : la visualisation, la structuration et l'affichage des données. Ils n'ont plus à trouver de structure générale pour le programme et les données puisqu'une forme familière leur est imposée.

L'intérêt du tableur est de permettre la résolution d'un grand nombre de problèmes et de simplifier certaines tâches (presque) sans programmation. Pouvoir traiter dans un même document des données numériques et textuelles, répond à la demande de millions d'employés du secteur tertiaire et de particuliers. Mais l'emploi d'un tableur nécessite une culture informatique que la majorité des utilisateurs actuelle a acquise petit à petit par la pratique. Si la maîtrise du tableur est un indicateur de compétence informatique, la manière dont les utilisateurs se l'approprient ne semble pas être une préoccupation majeure. Pour les éducateurs et les formateurs, il reste à identifier et énoncer précisément les compétences qui constituent les bases nécessaires à cette appropriation.

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. Au plan éducatif, les tableaux occupent une place notable dans l'éducation dès la maternelle. 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 leur place actuelle dans l'éducation.

¹⁰¹ <http://www.openoffice.org/> et <http://fr.openoffice.org/>

¹⁰² http://www.openoffice.org/press/ibm_press_release.html

¹⁰³ Dessins du 7, 8 et 9 août 2007

Consultés le 16 août 2007 <http://www.dilbert.com/comics/dilbert/archive/dilbert-20070807.html>

En outre, des formes automatisées d'agrégation de données en ligne sont dorénavant possibles à une large échelle et les offres gratuites de tableurs en ligne ouvrent la voie à des phénomènes de démocratisation similaires à ce que l'on a pu observer dans les années quatre-vingt. Pour certains philosophes, la *raison computationnelle* succède à la *raison graphique* (voir par exemple Bachimont, 2006). Une instrumentation autour des tables via le réseau mondial peut avoir des effets cognitifs équivalents à ceux mis en évidence par Jack Goody (op.cit.) à propos de l'écriture.

Ne devient-il pas urgent d'étudier la place que peut avoir la manipulation interactive de tableaux dans l'éducation ?

VI. Bibliographie

- Auray Nicolas (2000). *Politique de l'informatique et de l'information. Les pionniers de la nouvelle frontière électronique*. Thèse de sociologie de l'EHESS,
<http://egsh.enst.fr/auray/Auray%20These.pdf>
- Bachimont Bruno (2006). Support de connaissance et intelligence collective : héritage et individuation technique. Leçon à la Quatrième école d'été du GDR TIC et Société du CNRS.
<http://gdrtics.u-paris10.fr/pdf/ecoles/sept2006/BACHIMONT.pdf>
- Ballinger Daniel, Biddle Robert and Noble James (2003). Spreadsheet Visualisation to Improve End-user Understanding, The Australian Symposium on Information Visualisation, Adelaïde,
<http://www.mcs.vuw.ac.nz/~elvis/db/publications/comp489/invis-SSeuu.pdf>
- Baker J.E., Sugden J.S. (2003). Spreadsheets in Education – The first 25 Years, in *eJournal of Spreadsheets in Education 1(1)*, p. 18-43.
<http://epublications.bond.edu.au/ejsie/vol1/iss1/2/>
- Beaudouin-Lafon Michel (2007). 40 ans d'interaction homme-machine : points de repère et perspectives, *Interstices*.
<http://interstices.info/histoire-ihm>.
- Boulicaut, J-F., Marcel, P., Pinet, F., & Rigotti C. (1998). Spreadsheet generation using rule-based specifications. Proceedings 6th Int. Workshop on Deductive Databases and Logic Programming DDLP'98, Manchester (UK), June 20, 1998, (P. Fraternali, U. Geske, C. Ruiz and D. Seipel, editors), pp. 59-70.
<http://liris.cnrs.fr/~jboulica/ddlp98.pdf>
- Brown Polly S., Gould John D. (1987). An experimental study of people creating spreadsheets. ACM Transactions on Information Systems (TOIS), Volume 5 Issue 3, ACM Press, p. 258–272.
- Campbell-Kelly M. (2001). Not Only Microsoft: The Maturing of the Personal Computer Software Industry, 1982-1995, *Business History Review* 75, pp. 103-145,
<http://www.dcs.warwick.ac.uk/~mck/Personal/CampbellKelly2001.pdf>
- Campbell-Kelly Martin (2003). *From Airline Reservations to Sonic the Hedgehog*, MIT. Trad. Française : *Une histoire de l'industrie du logiciel. Des réservations aériennes à Sonic le Hérisson*, Vuibert Informatique.
- Campbell-Kelly Martin (2005). Not All Bad: An Historical Perspective on Software Patents, 11 *Mich. Telecomm. Tech. L. Rev.* 191,
<http://www.mttl.org/voleleven/campbell-kelly.pdf>
- Campbell-Kelly Martin, Croarken Mary, Flood Raymond, and Robson Eleanor (eds.) (2003). The history of mathematical tables: from Sumer to spreadsheets, Oxford University Press, 376 p.
- Capponi B., Balacheff N. (1989). Tableur et calcul algébrique. *Educational Studies in Mathematics* 20. 179-210,
- Cilloni Andrea (2005). La genesi della contabilità matriciale e la "ragioneria scientifica" del secolo decimonono (the roots of matrix accounting and the xix century "accounting science"), DE COMPUTIS (Spanish Journal of Accounting History), n° 2, juin 2005,
http://www.decomputis.org/dc/articulos_doctrinales/cilloni2.pdf
- Cilloni Andrea (2007). Spreadsheet Development in XIX Century European Matrix Accounting. 30th annual congress, European Accounting Association, Lisbon, 25-27 April 2007,
http://www.licom.pt/ea2007/papers/EAA2007_0095_final.pdf
- Cilloni Andrea, Marinoni Marco Angelo (2006). Spreadsheet, Chessboard and Matrix Accounting: The origin and development of advanced accounting instruments, 10th World Congress of Accounting Educators - WCAE 2006,
http://www.wcaestanbul2006.org/WCAE_ABSTRACTBOOK.pdf

- Clermont Markus (2003). A scalable Approach to Spreadsheet Visualisation. PHD dissertation. Universität Klagenfurt,
<http://www.isys.uni-klu.ac.at/PDF/2003-0175-MC.pdf>
- Croll Grenville J. The Importance and Criticality of Spreadsheets in the City of London, EuSpRIG 2005,
<http://www.eusprig.org/tiacositcol4.pdf>
- Fabre-Nicolin Raymond (1988). Le tableur et l'option informatique : vers la programmation par objet. Colloque francophone sur la didactique de l'informatique. EPI, p. 283-297.
- Goody Jack (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.
- Harrison Warren (2004). The Dangers of End-User Programming, IEEE Software ,
http://www.computer.org/portal/cms_docs_software/software/content/danger.pdf
- Ichbiah, D. (1993). *La grande aventure de Microsoft*. Marabout : Alleur. 315 p.
- Igarashi Takeo, Mackinlay Jock D., Chang Bay W., Zellweger Polle T. (1998). Fluid Visualization for Spreadsheet Structures, In Proceedings of 14th IEEE Symposium on Visual Languages , Halifax, Nova Scotia, Canada , pp.118-125,
<http://www-ui.is.s.u-tokyo.ac.jp/~takeo/papers/vl98.pdf>
- Leong Thin-Yin (2007). Simpler Spreadsheet Simulation of Multi-Server Queues, *INFORMS Transactions on Education*, Vol. 7, No 2, <http://ite.pubs.informs.org/Vol7No2/Leong/>
- Liebowitz Stan J. Margolis Stephen E. (1999). Winners, Losers, and Microsoft: How Technology Markets Choose Products. Chapter 8 Major Markets--WordProcessors and Spreadsheets.
<http://wwwpub.utdallas.edu/~liebowit/book/sheets/sheet.html>
- Mattessich R., Galassi G., (2000), "History of the spreadsheet: from matrix accounting to budget simulation and computerization", *Accounting and History. A selection of papers presented at the 8th World Congress of Accounting Historians, Madrid*, pp. 204-32. Traduzione spagnola "Historia de la hoja de calculo", *Rivista Internacional Legis de Contabilidad & Auditoria*, 18, pp. 41-86.
<http://www.mgeu.uma.es/CONFERENCIAS/Spreadsheet-Malaga-LV.pdf>
- Mattessich Richard (2005). A Concise History of Analytical Accounting: Examining the Use of Mathematical Notions in our Discipline. *DE COMPUTIS (Spanish Journal of Accounting History)*, n° 2, juin 2005,
http://www.decomputis.org/dc/articulos_doctrinales/mattessich2.pdf
- Mattessich Richard (2006). Foundational Research in Accounting: Professional Memoirs and Beyond. *DE COMPUTIS (Spanish Journal of Accounting History)*, n° 5, décembre 2006,
http://www.decomputis.org/dc/Mattessich_foundational_research_in_accounting.pdf
- Mattessich Richard (2006). La diversité des courants de recherché en comptabilité, évaluation et représentation. *Cahiers électroniques du CRECCI, IAE, Cahier 17*,
http://www.iae-bordeaux.fr/documents/cahiers_recherche/CahierCR17.pdf
- Nardi Bonnie A., Miller James R. (1990a). An ethnographic study of distributed problem solving in spreadsheet development, (*Proceedings of CSCW'90, 1990, 197-208*),
<http://www.miramontes.com/writing/ethno-spreadsheet/>
- Nardi Bonnie A., Miller James R. (1990b). The spreadsheet interface: A basis for end-user programming, (In D. Diaper et al (Eds.), *Human-Computer Interaction: INTERACT '90*. Amsterdam: North-Holland, 1990,
<http://www.miramontes.com/writing/spreadsheet-eup/>
- Panko Raymond (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, July 17-18, 2000. [En ligne]*
http://panko.shidler.hawaii.edu/SSR/Mypapers/EUSPRIG_2000.htm
- Piersol Kurt W. (1986). Object-oriented spreadsheets: the analytic spreadsheet package. *Conference proceedings on Object-oriented programming systems, languages and applications OOPLSA '86, Volume 21 Issue 11*, p. 385-390.
- Piguet Christian, Hügli Heinz (2004). *Du Zéro à l'ordinateur. Une brève histoire du calcul*. Presses polytechniques et universitaires romandes. 183 p.
- Plane Donald R. (1983). IFPS: A lot more than a spreadsheet. *Proceedings of the 15th conference on Winter Simulation - Volume 2 WSC '83, IEEE Press*, p. 605-608.
- Power Dan J. (2004). *A Brief History of Spreadsheets*, DSSResources.COM, World Wide Web,
<http://dssresources.com/history/sshistory.html>, version 3.6, 08/30/2004. Photo added September 24, 2002. Consulté le 26 juillet 2007.

- Reese Richard M. (1984). Simulation programming using spreadsheet software. Proceedings of the 16th conference on Winter simulation WSC '84, IEEE Press, p. 538–542.
- Robin Abraham (2007). End-User Software Engineering in the Spreadsheet Paradigm, PHD dissertation, Oregon State University, 219 p.,
<http://ir.library.oregonstate.edu/dspace/bitstream/1957/4953/1/thesis.pdf>
- 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, p. 207-214, 2005,
http://www.cs.cmu.edu/~cscaffid/papers/eu_20050923_vlhcc.pdf
- Shinners-Kennedy, D. (1986). Using Spreadsheets to Teach Computer Science. Proceedings of the seventeenth SIGCSE technical symposium on Computer science education, Cincinnati, Ohio, United States, ACM Press, p. 264-270
- Soper Jean B.; Lee Martin P. (1985). Spreadsheets in Teaching Statistics, *The Statistician*, Vol. 34, No. 3. (1985), p. 317-321.
- Thiriez, H. (2004). Spreadsheet-Based Professional Modelling *INFORMS Transactions on Education*, Vol. 4, No 2, <http://ite.pubs.informs.org/Vol4No2/Thiriez/>
- Walkenbach John (2004). Excel 2003 Power Programming With VBA (Broché). Chapter 1. Excel 2003: Where It Came From.
http://media.wiley.com/product_data/excerpt/26/07645407/0764540726.pdf
- Weiss H.J. and Rasmussen R.A. (2007). Lessons from Modeling Sudoku in Excel *INFORMS Transactions on Education*, Vol. 7, No 2,
<http://ite.pubs.informs.org/Vol7No2/WeissRasmussen/>

Annexe I : recherche de précurseurs à *VisiCalc* à partir de Wikipédia

Avec la requête « APLDOT spreadsheet », le 25 juillet 2007, Google fournit 136 pages qui se réduisent à une dizaine de propositions, compte tenu des doublons repérés. Mais toutes les pages proposées ne sont que des copies de l'article de Wikipedia.

Toutes citent correctement leur source, avec très souvent la mention : *This article may need to be rewritten. Please discuss this issue on the talk page.* (Cette dernière étant une copie de celle de Wikipedia, mais sans la possibilité d'interagir !). En fait, dans la plupart des cas, le contenu de ces pages est automatiquement mis à jour. Si on exclut la mention Wikipedia, Google ne trouve plus rien.

Première page de résultats

http://www.netformule.com/Cell_reference/encyclopedia.htm

<http://webraindor.info/wiki/spreadsheet>

<http://www.kerala.com/wiki-Spreadsheet>

<http://9855-spreadsheet.55.leahcology.com/>

<http://www.britain.tv/wikipedia.php?title=Spreadsheet>

http://www.myinfo24.com/index.php?vc=2&visilex_key=Spreadsheet.html

<http://www.encyclopediajr.com/wikiarticle/s/p/r/spreadsheet.php>

<http://www.inglesnoexterior.com/dictionary.aspx?q=Spreadsheet>

Deuxième page de résultats

http://zenbulogy.com/pages/cell_reference

http://www.liveinformation.org/live/live-photos-live-feeds-information-Cell_reference.html

Mais Google s'arrête à la seconde page.

Plus, une relance avec CSSTAB spreadsheet

http://www.joox.se/tellmemoreabout/index.php?option=com_content&task=view&id=175&article=Spreadsheet

<http://infoism.net/wiki/Spreadsheet>

http://www.mbceo.com/science/more_information.php?c=Spreadsheet

<http://adsense-success-guide.com/wiki/Spreadsheet>

http://suche.maxic.to/lexikon_Spreadsheet_en.html

<http://golf-golfers.com/index.php?q=Spreadsheet>

<http://www.ebamgoo.com/?q=Spreadsheet>

Annexe II : controverse autour de l'article Spreadsheet de Wikipedia

La description de *The Works Records Systems (WRS)*, censé être conçu par Robert Mais dans la société ICI (Imperial Chemical Industries) et implémenté par une équipe comprenant un certain Ken Dakin (qui semble être à l'origine des informations) a été supprimée de la version du 29 juillet de Wikipedia, qui conserve cependant une note interne :

One example cited recently in Wikipedia is the CICS-based Works Records System built by Imperial Chemical Industries in the early 1970s. Other innovative approaches to end-user computing were being pursued at Xerox PARC, MIT, Citibank, National CSS, and IBM.

L'article *Imperial Chemical Industries* contient lui une partie consacrée au *Works Records System*. Cette partie est précédée du bandeau : "*This article or section may contain original research or unverified claims*".

La mention d'un autre système *Autoplan/Autotab* est sans doute indirectement liée à *WRS*. En effet, une recherche à l'aide de Google ne fournit rien sur *Autoplan/Autotab* si ce n'est une réflexion dans un blog, celui de Creative Karma (Doug Pardee). Dans un billet intitulé *The First Computerized Spreadsheet Program*¹⁰⁴, daté du 2 décembre 2006, il fournit un texte qui va se retrouver mot pour mot dans une version suivante de Wikipedia.

En effet, la version du 2 décembre contient un développement autour de *Works Records System*, rien sur *Autoplan/Autotab*. Cette partie est ajoutée le 15 décembre 2006. Et on retrouve dans les discussions sur l'article (onglet discussion de l'article *Spreadsheet* de Wikipedia), une intervention de l'auteur du blog :

"The text for this section was lifted (with minor changes) from my weblog without permission. Nevertheless, I'll let it stand. I hereby grant Wikipedia permission to use that section of text.

I would note, however, that my weblog article was written entirely from my personal remembrances and IMO is not up to the "reliable source" standards of Wikipedia. If it was, I would have included it in the Wikipedia article myself."

Doug Pardee 22:51, 2 January 2007

Un commentaire était ajouté à l'article proposé par Doug Pardee sur Creative Karma :

"I can concur with the fact that Visicalc was not the first spreadsheet or even interactive spreadsheet as I worked on the first one known as the "Works Records System" at ICI in the UK around 1974 (six years before Bricklin). I have described the history of Autoplan on the Wikipedia article on "spreadsheet" with reference to your article and a link to the Creative Karma website.

I knew Leroy Ellison personally but didn't know about his earlier work on Autoplan and he didn't know about my work 6/7 years earlier on interactive spreadsheets either!

Capex bought some test/debug technology from me to produce their "EZtest CICS" product to compliment their Capex Optimizer for COBOL."

Posted by: ken dakin on Friday, 15 December 2006¹⁰⁵

Le même Ken Dakin va ajouter un nouveau commentaire le 14 août 2007, suite à ses démêlés avec Wikipedia. En tout cas, de l'aveu même du contributeur (Pardee), la référence est faible et cela conduit à douter du bien fondé des précisions données dans l'article *Spreadsheet*. On pourrait faire l'hypothèse que Ken Dakin a ajouté le texte sur *Autoplan/Autotab* (toujours présent dans la version de l'article de fin juillet) afin de mieux faire passer les

¹⁰⁴ http://creativekarma.com/ee.php/weblog/comments/the_first_computerized_spreadsheet_program/

¹⁰⁵ http://creativekarma.com/ee.php/weblog/comments/the_first_computerized_spreadsheet_program/

développements sur *Works Records Systems*, ce qui va conduire à une polémique dont on trouve la trace dans la page de discussion de l'article de Wikipedia.

En effet, une longue discussion s'est tenue en juin 2007 entre Dan Bricklin, qui a mis des commentaires sur son blog¹⁰⁶ (le mercredi 6 juin 2007) et critique notamment la place trop importante accordée à *Works Records Systems* (WRS) et celui qui a introduit les éléments sur ce système¹⁰⁷. On note une intervention non signée du 12 juin 2007, mais attribuée à Ken Dakin. On apprend que l'article IBM 3270 contient également la référence à WRS¹⁰⁸. La discussion porte essentiellement sur le manque de sources attestant de ce que pouvait être WRS et l'impossibilité de vérifier les affirmations de Ken Dakin.

Trevor Hanson, un responsable wikipédien, explique (le 20 juillet 2007) qu'il va retirer les développements sur WRS et les réintroduire en partie dans l'article présentant ICI (Imperial Chemical Industries), après un appel à d'autres avis (resté lettre morte).

On constate deux dernières interventions de Ken Dakin les 14 et 15 août, précisant qu'il a numérisé et mis en ligne une partie de la documentation de WRS¹⁰⁹, et signale également le blog créé le 25 juin 2007 pour défendre ICI Works Records Systems comme tableur interactif existant avant *VisiCalc*¹¹⁰. Il conclut :

In view of the importance of Spreadsheets to the world and in the name of historical truth, I have suggested to the "museum of computing" that they try to obtain the original source code direct from ICI by appealing to the ICI directors to search their archives for copies of it (it was all held on "Panvalet" libraries which had archiving features). ken 04:02, 15 August 2007 (UTC)

Le débat n'est certainement pas encore clos.

¹⁰⁶ http://danbricklin.com/log/2007_04_12.htm#spreadsheetpage

¹⁰⁷ <http://en.wikipedia.org/wiki/Talk:Spreadsheet>

¹⁰⁸ <http://en.wikipedia.org/wiki/3270>

¹⁰⁹ http://www.editgrid.com/user/ken/WORKS_RECORDS_SYSTEM

¹¹⁰ <http://worksrecordssystem.blogspot.com/>

Annexe III : Une partie de l'offre tableur en avril 1982

Compagnie	Nom/prix	Configuration/capacité
Addison Wesley Publishing	MICRO DSS/Finance \$1500	Apple II, 48k and Pascal.
Comshare Target Software (formerly AMSI)	Planner Calc \$195	CP/M based, runs with Apple DOS and MS DOS.
	Target Planner \$195	CP/M based, 48k minimum.
Context Management Systems	M.B.A. \$695	IBM and Apple III; package contains electronic spreadsheet, data-base management, word processing, communications, graphics.
Cromemco	PlanMaster \$295	Cromemco systems only.
Hayden Publishing	FINPLAN tape \$69.96, disk \$74.95	TRS-80 Models only.
Intelligent Systems	COLOR CALC \$200	Intelligent Systems 3650 in extended BASIC only.
Lifeboat Associates	FPL \$745	Runs on CP/M 80 or MPM; 2 versions, 48k and 56k; one disk drive with 125k or two drives with 175k each; must have clear screen and cursor addressing.
	T/Maker 2 \$275	Runs on CRT with clear screen and cursor addressing, or on CDOS; 48k minimum.
MicroPro International	CALCSTAR \$295	Runs on CP/M version 2.0 or above with 80-column screen, addressable cursor and 48k minimum; 56k recommended.
Ohio Scientific	PLANNER PLUS \$149	Operates on hard disk C3B 74 megabyte, C3C 23 megabyte, and C3D 7 megabyte only.
Osborne McGraw-Hill	MICROFINESS \$495	Runs on Apple II; is a UCSD program; has target search and sensitivity analysis.
Peachtree Software	MAGICALC \$300	CP/M based; 44k minimum.
Radio Shack	SPECTACULATOR \$39.95	Runs on Color Computer.
Software Products	LOGICALC \$290	Runs on Z80 machines or those that support UCSD Pascal, 56k.
Sorcim	SUPERCALC \$295	Runs on CP/M-based machines.
Structured Systems Group	MAGIC WORKSHEET \$250	8080 and Z80 based; runs on terminals that support 48k minimum.
Vector Graphic	EXECUPLAN \$150	Vector Graphic systems only.
Visicorp (formerly Personal Software)	DESKTOP PLAN II \$250	Runs on Apple II and III.
	DESKTOP PLAN III \$250	48k minimum; Apple II and III, IBM Personal Computer, Atari, Commodore PET, TRS-80 Models I, II and III, and HP 125.
	VISICALC \$250	Apple II and III, IBM Personal Computer, Atari 8000, Commodore 8032, 32k PET, TRS-80 Models I, II and III, HP 85, HP 87, HP 125.
Westico	MINIMODEL \$495	Runs on most CP/M-based machines with 48k minimum; two disk drives; also runs with CBASIC.

Scenarios For Success: The Vision Of Spreadsheets¹¹¹
(repris de Personal Computing, issue 4/1982, pp. 58-63)

¹¹¹ <http://www.aresluna.org/attached/computerhistory/articles/spreadsheets/scenariosforsuccess>

Annexe IV : Une partie de l'offre tableur en été 1987

Table 1: Spreadsheet prices and company information

Spreadsheet	Versions	Hardware	Company	List price	Manual(s)
1-2-3	2.01	IBM PC; 256K	Lotus Development Corp.	\$495	551 pages
HAL	2.01	IBM PC; 512K	Lotus Development Corp.	\$150	178 pages
Boeing CALC	3.0	IBM XT, AT; 386K and hard disk	Boeing Computer Services Microsoft products	\$399; LAN: \$799	380 pages
Excel	1.03	Mac; 512K	Microsoft Corp.	\$395	572 pages
Multiplan	3.01	IBM PC; 256K	Microsoft Corp.	\$195	516 pages
Silk	1.0	IBM PC; 512K	Daybreak Technologies	\$298	292 pages
SuperCalc4	1.0	IBM PC; 256K	Computer Associates Int'l Inc.	\$495	650 pages
Trapeze	1.0	Mac; 512K	Data Tailor	\$295	208 pages
VP-Planner	1.3	IBM PC; 256K	Paperback Software International	\$99.95	500 pages

Table 2: Features of the nine spreadsheets reviewed

Feature	1-2-3	1-2-3/ HAL	Boeing CALC	Excel	Multiplan	Silk	SuperCalc4	Trapeze	VP- Plan
Computer	IBM PC	IBM PC	IBM PC	Mac	IBM PC	IBM PC	IBM PC	Mac	IBM PC
Functions	89	89	89	114	77	81	92	127	59
Macro commands	42	349	0	29	13	44	46	0	9
Networking	No	No	Yes	No	P	No	No	No	No
Arithmetic coprocessor	Yes	Yes	No	Yes	No	Yes	Yes	Yes	No
Expanded memory	Yes	Yes	*	N/A	No	Yes	Yes	N/A	No
Multidimensional	No	No	Yes	No	No	No	No	No	Yes
Linkage	P	Yes	Yes	Yes	Yes	P	P	Yes	P
Database	Yes	Yes	No	Yes	P	Yes	Yes	No	Yes
Natural language	No	Yes	No	No	No	No	No	No	No
Undo	No	Yes	No	Yes	No	No	No	Yes	No
Record macros	No	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
Contextual help	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
Live help	No	No	No	No	No	Yes	No	No	No
Auditing	P	Yes	No	Yes	Yes	Yes	P	No	P
Graphics	Yes	Yes	No	Yes	No	Yes	Yes	Yes	Yes
Cross-compatibility	No	No	Yes	Yes	Yes	Yes	Yes	No	Yes
Non-copy-protected	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

This new crop of advanced programs offers multidimensionality and natural language¹¹², Rich Malloy
(Reprinted from Byte Summer 1987 issue, pp. 69-75)

¹¹² <http://www.aresluna.org/attached/computerhistory/articles/spreadsheets/newcrop>