

Tableurs, risques et évolution technologique
Compte rendu de la présentation du 8 janvier 2019, à la Villa Violet

Compte rendu rédigé par Laure MUSELLI & ANDSI

En bref...

Tie CHENG, spécialiste des tableurs et fondateur de la société Matrix Lead, partant du constat qu'Excel constitue aujourd'hui un outil irremplaçable dans les entreprises, propose des solutions permettant de réduire les fragilités inhérentes à cet outil. Après avoir rappelé ce qui fait d'Excel un langage de programmation fonctionnel, il présente des outils permettant de rendre les tableurs plus sûrs, efficaces et puissants. Pour finir, il adopte une vision plus stratégique et passe en revue les différentes technologies présentes dans l'entreprise, afin de donner quelques pistes sur la façon de les associer à Excel.

L'Association Nationale des Directeurs des Systèmes d'Information organise des débats et en diffuse des comptes-rendus, les idées restant de la seule responsabilité de leurs auteurs. Elle peut également diffuser les commentaires que suscitent ces documents.

Excel : un outil irremplaçable mais source de polémiques

Excel est aujourd'hui un outil omniprésent en entreprise et irremplaçable car utilisé par tous les métiers, qui peuvent passer un temps considérable à développer des tables de grande taille et d'une grande complexité.

Les DSI présents lors de la réunion relèvent en effet que :

- Excel, s'il est beaucoup utilisé, l'est en silos, afin de résoudre des problématiques individuelles, mais sans possibilité de collaboration ;
- Les erreurs sont considérées comme inhérentes aux outils développés sous Excel et de fait acceptées ;
- Il arrive que des collaborateurs de la DSI aillent puiser des données dans les tableurs développés individuellement par certains collaborateurs pour les diffuser à l'ensemble de l'entreprise.

Toutefois, de par leur complexité, les tableurs constituent une source d'erreurs difficiles à contrôler. Des solutions sont donc nécessaires pour simplifier le travail des utilisateurs de ces outils, augmenter leur efficacité, et in fine les guider vers le monde de la programmation.

Excel pouvant être considéré comme un langage de programmation, voire un environnement de programmation, s'inspirer de notions, principes et outils d'autres langages pourrait donc permettre de mieux programmer et gérer Excel, afin de développer des outils en interne.

Excel, un langage de programmation fonctionnel

Si Excel est généralement appréhendé comme un **outil permettant de manipuler des données**, en réalisant principalement des copier-coller, il peut être également considéré, du point de vue de la programmation, comme :

- Un **outil permettant une semi-automatisation**, grâce aux différents boutons, menus et add-ins disponibles ;
- Un **outil d'automatisation de tâches**, grâce aux fonctions, formules et à VBA, c'est-à-dire un **outil de programmation** ;
- Un **outil de programmation fonctionnel**, dans lequel **les formules constituent les fonctions et les chaînes de formules constituent un programme Excel** (tableur). Les programmes VBA, eux, permettent des programmations plus complexes, notamment orientées objet. L'évaluation d'un programme Excel consiste donc à évaluer les formules dans les chaînes une par une.

La programmation fonctionnelle possède des avantages indéniables sur la programmation par procédures :

- **Elle ne produit pas d'effets de bord**, c'est-à-dire n'a pas d'autres effets que la tâche programmée, comme la modification de valeur de variables globales. En revanche, dans une macro VBA mal programmée, ces effets de bord peuvent être présents ;
- Les **dépendances** (relations entre variables et programmes) **sont claires et s'adaptent aux changements effectués dans Excel** (en cas d'ajout de colonnes, les formules sont mises à jour). En revanche, lorsque l'on programme en macros VBA, la dépendance est difficile à trouver (les entrées et sorties sont moins évidentes qu'en Excel) et elle reste figée, car ne s'adaptant pas aux changements effectués dans la feuille Excel, ce qui peut générer des erreurs.

Excel étant par essence un langage fonctionnel, respecter ce paradigme est donc préférable.

S'inspirer de notions, principes et outils d'autres langages pour mieux programmer en Excel

Finalement :

- Excel n'est plus un logiciel, mais un langage et un outil de programmation ;
- Les tableurs ne sont plus une collection de données, mais un programme ;
- Les utilisateurs d'Excel ont désormais une casquette de « développeurs ».

Dès lors, de quels principes, notions et outils d'autres langages peut-on s'inspirer ? Que peut-on appliquer à Excel, afin de créer des tableurs plus sûrs, efficaces et puissants ?

Vérification des programmes par les types

Excel et VBA sont des langages faiblement typés, dans le sens où :

- Les données de différents types (numériques, alphanumériques, dates, etc.) se convertissent rapidement ;
- Les règles de conversion de type ne sont pas claires ;
- Les opérations de types qui n'ont pas de sens ne génèrent pas toujours d'erreur et ne sont donc pas bloquantes, ce qui fait des tableurs une source d'erreurs cachées difficiles à contrôler.

Des outils de vérification des programmes par les types permettraient de détecter des erreurs en identifiant les incohérences de types avant l'exécution d'un programme. Cette méthode de vérification, qui existe dans de nombreux langages comme OCaml et Haskell, s'avère très efficace, car elle permet de repérer environ 80% des erreurs d'un programme, mais n'existe pas dans Excel et VBA.

Renforcer les types dans Excel VBA et appliquer un système de programmation fortement typé peut permettre de détecter les erreurs. Les JavaScript add-ins et les Office add-ins développés par Microsoft mais aussi par des développeurs tiers comme Matrix Lead constituent des outils permettant ce type de vérification.

L'outil Spreadsheet Verificator permet par exemple d'identifier de potentielles erreurs, comme des cellules non-remplies ou mal saisies. Les erreurs de saisie se répercutent en cascade et peuvent ainsi causer d'autres erreurs de type, qui sont autant d'indices pour trouver les erreurs de saisie.

Outils d'inspection des mauvaises pratiques

Les risques liés aux tableurs ne concernent pas que les erreurs, mais aussi l'inefficacité des programmes, qui peut être la conséquence de mauvaises pratiques concernant :

- **Les calculs**, qui peuvent être répétitifs au niveau de la structure de la formule ou au sein d'une même cellule, impliquer des fonctions volatiles qui demandent de recalculer l'ensemble des feuilles ou même des formules matricielles, qui mettent du temps à se calculer ;
- **Les formats**, avec des styles inutilisés ou des mises en forme conditionnelles ;
- **Les dépendances dans les formules VBA**, qui peuvent faire planter le programme.

Matrix Lead propose différents outils d'inspection de mauvaises pratiques. Bottleneck Detector, développé en JavaScript et HTML et connecté à un analyseur de noyau développé en OCaml, permet de rechercher les cellules qui ralentissent la feuille, afin d'améliorer le temps de réponse.

D'autres outils d'inspection permettent d'uploader une feuille de calcul et de recevoir un rapport faisant l'inventaire des dépendances entre les macros VBA et les cellules et feuilles. Les macros inutiles ou critiques sont détectées, ainsi que les entrées et sorties importantes.

Editeurs de formules

Certains développeurs sont habitués à d'autres langages comme JavaScript ou C++. Des éditeurs existent, qui permettent d'aider les développeurs à mieux « développer » sous Excel, en rendant les formules plus compréhensibles, c'est-à-dire en les scindant en plusieurs lignes, en ajoutant des indentations pour les rendre plus lisibles ou en colorant les fonctions clés. Formula Formatter est l'éditeur développé par Matrix Lead.

Stratégie : comment associer les différentes technologies de l'entreprise à Excel ?

Il est courant de développer des bibliothèques en **C++ ou C#**, ensuite emballées dans des DLL. Excel et VBA appellent ensuite ces bibliothèques au sein du tableur pour écrire les « user-defined functions ».

La technologie **VSTO (Visual Studio Tools for Office)** permet également d'écrire les « user-defined functions », mais aussi de créer les interfaces dans Excel.

Python, qui est un langage très à la mode, peut également être connecté à Excel. Chez JP Morgan et Merrill Lynch, il existe des plateformes de programmation en Python, qui centralisent les programmes en Python et écrivent des bibliothèques que les utilisateurs d'Excel peuvent appeler. Chez Pimco, un gérant de fonds américain, un éditeur de Python a été intégré dans Excel. Il est donc possible de coder en Python et utiliser ensuite les fonctions dans Excel. Tout comme dans GitHub, on peut uploader un repository et charger vers un repository, afin que tout le monde travaille sur les mêmes fonctions.

Les add-ins **JavaScript apps for Office** sont disponibles dans le menu de la version 2013 d'Office, avec une version plus aboutie dans Office 2016. Ce sont des outils développés en langage web tels HTML, JavaScript, CSS, etc., qui sont disponibles sur une boutique en ligne.

L'outil Funfun est destiné aux utilisateurs qui n'ont pas leur propre serveur. Il permet de créer des **programmes JavaScript directement dans Excel**. Un éditeur JavaScript permet, à partir des données d'entrée et des programmes, d'avoir en sortie des données qui changent en fonction des modifications effectuées sur ces deux éléments. Les bibliothèques très puissantes d'analyse de données et de visualisation de JavaScript peuvent donc être mobilisées par ce biais. Il est également possible de charger directement sur Excel, à partir d'un site web développé par Matrix Lead, de nombreux programmes d'analyse de données qui sont modifiables.

Les avantages et inconvénients de chacune de ces technologies sont les suivants :

- **VBA** est souvent attaché à chaque tableur, décentralisé, et il est donc difficile de contrôler la qualité et de partager avec les équipes ;
- **VSTO** est un outil mature et assez puissant, et qui permet d'accéder au PC ;
- **Python** ne possède pas vraiment d'interface et permet plutôt de développer les « user-defined functions », c'est-à-dire les fonctions de calcul. Il faut faire appel à d'autres outils comme VSTO et VBA pour développer des interfaces utilisateurs ; en revanche, il s'agit d'un langage très populaire pour tout ce qui concerne l'analyse de données et la visualisation ;
- **JavaScript** a l'avantage de posséder une interface ergonomique, une boutique en ligne Microsoft sur laquelle sont disponibles des outils, un web illimité, et des outils d'analyse de données puissants comme TensorFlow. Il possède par ailleurs l'appui de Microsoft.

Débat

Intervenant : Que pensez-vous de power BI ? Comment cet outil peut-il consolider de l'Excel ?

Tie CHENG : Je le considère comme un logiciel Microsoft de Business Intelligence comme les autres. Power BI est un outil qui permet de gérer une grande quantité de données, alors que dans Excel, lorsque la quantité de données est trop importante, il faut passer à un outil plus spécialisé.

Int. : Avez-vous entendu parler de l'outil Anaplan ? Il s'agit d'un tableur dans le cloud centralisé et versionné. Il possède un intérêt en ce qui concerne la centralisation et la traçabilité des données.

T. C. : Non, pas vraiment.

Int. : Anaplan est un outil assez à la mode utilisé en in memory dans le cloud, notamment pour les processus de planification budgétaire et le collaboratif.

Int. : Avec votre expérience, quelles sont les limites raisonnables d'Excel ? Car tout le monde l'utilise pour tout et n'importe quoi...

T. C. : Aujourd'hui, on a des millions de lignes dans Excel, mais il n'est pas raisonnable d'avoir des millions de données. Je pense que quelques milliers de lignes de données est un nombre gérable et raisonnable sous Excel.

Int. : Il s'agit de la plus belle invention informatique, mais cela dépend de la personne qui est derrière, car on peut faire des choses formidables en programmation. Quelles sont les limites environnementales ? C'est peut-être plus approprié en monoposte, plutôt que pour 4 000 personnes en centralisé...

T. C. : Dans les entreprises, je pense que l'on peut avoir des front-ends comme Excel avec d'autres plug-ins. En revanche, le back-end est centralisé.

T. C. : Parmi vous, certains ont-ils eu un usage avancé d'Excel tel que je l'ai décrit ? Avez-vous des plug-ins dans Excel ?

Int : Celui de statistiques...

Int. : Dans une de mes vies antérieures, j'ai utilisé Excel à un niveau très avancé, comme partie intégrante du système d'information dans un institut de recherche, où l'on faisait de l'analyse de variance, de l'analyse en composantes principales et de l'analyse factorielle des correspondances. Il existait des modules de très haut niveau, mais cela est adapté plutôt à un usage monoposte pour le calcul, avec quelques données en back-end, et là ça devient puissant et pertinent. Mais si on veut brancher 15 000 utilisateurs, ce n'est pas adapté, car ce n'est pas collaboratif. Il faudra donc aller chercher vers du Power BI, mais on fera beaucoup moins de choses sur Power BI que sur Excel, où l'on peut programmer en Python. Ensuite, il faut normaliser et verrouiller les feuilles de calcul.

Int. : Mais la difficulté, est-ce que ça n'est pas l'évolutivité, avec les différentes versions ?

Int. : Il existe une tierce maintenance applicative et des développeurs qui entretiennent cela comme un véritable ERP.

Int. : En général, lorsqu'on passe d'Excel vers une application, c'est que l'on recherche du collaboratif et quelque chose de plus orienté process que données.

Int. : Que pensez-vous de Microsoft Excel Online ?

T. C. : Je trouve cela pas mal du tout. L'avantage de JavaScript apps for Office est qu'il est cross-platform : Mac, mobile, online et PC.

Présentation de l'orateur

Tie CHENG a effectué une thèse en analyse statique d'applications tableurs chez Inria & ENS Paris & Ecole Polytechnique. Il est actuellement le CEO de Matrix Lead (www.matrixlead.com), une startup qui fournit aux professionnels et entreprises les technologies et solutions avancées de tableurs....