N° d'ordre :
N° de série :

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE ECHAHID HAMMA LAKHDAR - EL OUED
FACULTÉ DES SCIENCES EXACTES
Département D'Informatique

Mémoire de Fin D'étude
Présenté pour l'obtention du Diplôme de

LICENCE ACADEMIQUE

Domaine : **Mathématique et Informatique**
Filière : **Informatique**
Spécialité : **Systèmes Informatiques**

Thème

Développement d'un outil de design des tableaux pour La LaTeX

Présenté par :

- **KOULL MOHAMMED FOUAD**
- **MAAMRA BADIE**
- **LABRECHE RIDHA**

Proposé et Encadré par : **M.ZAIZ FAOUZI**

Soutenue le 20/05/ 2018, Devant le jury:

M. **KHELAIFA Abdennacer** Examinateur.

M. **NAOUI Mohammed Anouar** Examinateur.

Année Universitaire: 2017-2018

Dédicaces

Tout d'abord, je remercie le bon Dieu, Clément et Miséricordieux pour son aide éternel.
Je dédie cette thèse, en premier lieu, à la mémoire de mon grand père Echeickh Ali MAAMRA: le grand savant qui a su comment résister seul face à une population d'ignorants afin qu'il puisse leur éclairer la voie de l'islam tel qu'il était interprété par les vrais savants (Assalaf Essaleh) et qui a pu nous inculquer l'apprentissage du coran.
Je dédie, par la suite, ce travail à ma chère mère pour sa tendresse et mon cher père pour sa patience et encouragement.
Aussi à mes très chers frères et mes chères sœurs pour leurs soutien,
A mes cousins et cousines,
A tous ceux que j'aime,
A tous mes amis.

MAAMRA BADIE

Dédicaces

Au nom d'Allah le plus grand merci lui revient de me avoir guider vers le droit chemin.
Je dédie ce modeste travail : à:
À mes très chers parents qui ont toujours été là pour moi, et qui m'ont donné un magnifique modèle de labeur et de persévérance
À ma petite famille , ma chère femme qui m'a soutenu, mes petit anges MARIA et SERINE.
À mes frères et mes sœurs, chacun à son nom ;
À toute la famille ;
À tous mes amis ;
À tous mes chers enseignants qui m'ont enseigné.

KOULL Mohammed Fouad

Dédicaces

J'ai le plaisir de dédier ce travail :
A DIEU le plus puissant Qui m'a donnée la santé, la force Le courage, la croyance, le soutien « malgré toute les difficultés »
A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études,
A mes chères sœurs pour leurs encouragements permanents, et leur soutien moral,
A mes chers frères pour leur appui et leur encouragement,
A toute ma famille pour leur soutien tout au long de mon parcours universitaire, A tous ceux qui m'ont donné de l'aide ...
A tous ceux qui m'ont donné des conseils..

LABRECHE Ridha

Remerciements

A Dieu, le tout puissant, nous rendons grâce pour nous avoir donné santé, patience, volonté et surtout raison.
On dit souvent que le trajet est aussi important que la destination
En premier lieu, nous aimerions remercier notre encadreur Monsieur ZAIZ FOUAZI qui nous a aidé et qui nous a donné des conseils durant ce travail.

Nous remercions tous les enseignants qui nous ont inculqué beaucoup de connaissances
nous remercions chaleureusement les membres du jury qui nous ont fait l'honneur de participer à notre jury et accepter de juger ce modeste travail.

En fin, nous voulons encore remercier tous ceux qui nous ont consolidé et nous ont encouragé entres autres les parents et les amis.

Résumé

LaTeX est un programme gratuit qui est l'un des meilleurs et des plus distribués dans le monde. Conçu par Leslie Lamport en 1982, il s'agit d'une extension du programme TeX conçu par Donald E. Knuth, professeur de mathématiques en 1978. La popularité de ce programme est due à sa présence dans le cadre d'Unix et Linux. Le programme LaTeX pour la rédaction et la coordination des documents scientifiques est plus que juste d'être un éditeur en raison du concept programmatique de ce programme exceptionnel, qui nécessite une logique de réflexion lorsque l'étudiant prépare le rapport. Néanmoins, l'inconvénient de ce programme, c'est la difficulté de l'utiliser. Par exemple, la création des tableaux de façon manuelle est difficile, surtout pour les débutants car ce n'est pas comme les programmes de traitement de texte basés sur WYSIWYG (ce que vous voyez ce que vous voyez est ce que vous obtenez.), L'écriture est ce qu'elle obtiendra après l'impression. C'est une fonctionnalité flexible qui favorise les traitements de texte modernes pour beaucoup d'utilisateurs, mais la facilité d'utilisation se fait au détriment d'un avantage important: (la formation et la préparation d'une structure claire du rapport, son début, détails et fin sont connus)
Ce travail, vise à développé un outil de dessin qui permet de créer des tableaux et de les modifier à l'aide des opérations, telles que: fusionner et diviser des cellules, ajouter des colonnes et des lignes, éditer du texte dans des tableaux et autres opérations. C'est dans le but d'accélérer la productivité de l'utilisateur et de simplifier l'automatisation de la génération du code grâce à une interface graphique simplifiée. Cet outil permet d'accéder au code LaTeX avec zéro erreur.

Mots clés: LaTeX, Tableau, PostScript.

Abstract

LaTeX is a free program that is one of the best and most widely distributed programs in the world. Designed by Leslie Lamport in 1982, it is an extension of the TeX program designed by Donald E. Knuth, a mathematics professor in 1978. The popularity of this program is due to its presence as part of Unix and Linux. LaTeX program for writing and coordinating scientific documents is more than right to be an editor because of the programmatic concept of this outstanding program, which requires logic in thinking as the student prepares the report. Creating tables manually is difficult, especially for beginners because it is not like word processing programs based on "WYSIWYG" what you see is what you get, which means .This is a flexible feature that favors modern word processors for many users, but the ease of use comes at the expense of an important advantage: (the formation and preparation of a clear structure of the report, its beginning, details and end are all known).

Through this work we have developed a drawing tool that allows creating tables and modifying them in various operations such as: merging and dividing cells, adding columns and lines, editing text in tables and other operations. This is in order to speed up the user's productivity and simplify the automation of the spreadsheet code through a simplified graphical interface. The tool allows access to LaTeXcode with zero error.

Keywords: LaTeX, Table, PostScript.

الملخص

برنامج LaTeX لكتابة وتنسيق الرسائل العلمية هو برنامج مجاني مميز، ويعد واحدا من أفضل برامج تنسيق الرسائل العلمية وأكثرها انتشارا في العالم. صممه ليزلي لامبورث سنة 1982 و هو امتداد لبرنامج TeX الذي صمم استاذ الرياضيات دونالد كانولث سنة 1978. ويعود الفضل في شهرة هذا البرنامج لوجوده كجزء من نظام يونكس ولينكس. كما يعد برنامج LaTeX لكتابة وتنسيق الرسائل العلمية مترجم أكثر من كونه محرر وذلك بسبب المفهوم البرمجي القائم عليه هذا البرنامج المميز والذي يتطلب المنطق في التفكير أثناء قيام الطالب بإعداد التقرير. لكن ما يعيب هذا البرنامج صعوبة استخدامه إنشاء الكود الخاص بالجداول يدويا أمر صعب خاصة بالنسبة للمبتدئين لأنه ليس كبرامج معالجة النصوص التي تعتمد على مبدأ WYSIWYG أي أن أي عملية يجريها المستخدم على النص سيرى أثرها جلياًا أمامه مباشرة و ما يراه من تأثيرات على النص وقت الكتابة هو ما سيحصل عليه بعد طباعته للتقرير . إلا أن السهولة في الاستخدام تأتي على حساب ميزة مهمة ألا وهي :« تشكيل و إعداد هيكلة واضحة للتقرير، بدايته و تفاصيله و نهايته كلها معروفة »

من خلال هذا العمل قمنا بتطوير أداة رسم تسمح بإنشاء جداول و التعديل عليها بمختلفة العمليات مثل: دمج خلايا و تقسيمها، إضافة أعمدة و أسطر، التعديل على النصوص داخل الجداول و غيرها من العمليات.و هذا من أجل تسريع مردودية المستعمل و تبسيط الكتابة الآلية للكود الخاص بالجداول من خلال واجهة رسومية مبسطة. الأداة تسمح بالحصول على كود خاص ببرنامج LaTeX مع نسبة خطأ معدومة.

كلمات مفتاحية: LaTeX ، **جدول**، PostScript.

Table des matières

Liste des figures

Liste des tableaux

Introduction

LaTeX (prononcé latec) est un formateur de texte spécialement orienté vers la création des documents scientifiques de haute qualité et qui produit des documents d'une excellente qualité et bien structurées. Il est constitué d'un ensemble de commandes TeX qui permettent à l'utilisateur une rédaction plus aisée. LaTeX n'utilise pas le principe de "WYSIWYG" ("What You See Is What You Get") comme Word, WordPerfect, Works ou d'autres. A l'heure du Web, on peut comparer LaTeX au "HTML". Pour composer un texte avec ce dernier langage, on crée à l'aide d'un éditeur de textes, un fichier contenant à la fois le texte et des commandes de mise en page et de mise en forme de caractères. En LaTeX, le principe est identique mais plus puissant. En effet, contrairement au HTML, il est possible, par exemple, de spécifier la taille des pages (largeur et hauteur du texte) et d'insérer aisément des formules mathématiques. LaTeXest même plus qu'un logiciel de traitement de texte, c'est un langage de programmation qui permet à l'utilisateur de définir ses propres commandes pour automatiser la mise en forme ou effectuer des tâches répétitives. Néanmoins, pour un débutant, il est très difficile de créer et manipuler manuellement le code de certains éléments tels que les tableaux (simple ou complexes), les diagrammes,... etc. Dans notre travail nous avons proposé un outil de génération du code des tableaux LaTeX . Cet outil permet à l'utilisateur de créer à l'aide d'une interface graphique des tableaux Latex des différents types, à savoir :

- Tableaux simples.
- Tableaux avec cases fusionnées horizontalement ou verticalement
- Tableaux avec des cases contenant un séparateur diagonal

Le reste du document est organisé comme suit :

Le premier chapitre présente un état de l'art sur Latex et différentes commandes de création des tableaux.

Le deuxième chapitre donne une conception et la mise en œuvre du système à développer.

Le troisième chapitre présente le langage de programmation choisi ainsi que l'utilisation de l'interface du l'application.

Nous terminons ce travail par une conclusion générale sur les résultats obtenus et quelques perspectives a ajouter à ce travail.

Chapitre 1

État de l'art

Introduction

LaTeX est un système puissant de production de documents, particulièrement utilisé dans certaines disciplines scientifiques, car il permet de générer des formules et expressions symboliques très complexes. Le fonctionnement de ce logiciel est basé sur un système de commandes, sur plusieurs applications et distributions complémentaires, dont l'utilisation varie selon les environnements informatiques.

Dans ce chapitre, nous allons voir un aperçu sur LaTeX ainsi qu'un petit survol sur les différentes méthodes de traitement des tableaux.

1.1 Les Logiciels Traitement des Texte

Un logiciel de traitement de texte permet d'utiliser un ordinateur pour rédiger, corriger et imprimer des documents écrits tels que des lettres, des articles de presse, des factures, des contrats ou de la publicité. Dans sa forme la plus simple, un programme permet de faire exactement ce qui est fait avec une machine à écrire. Des programmes plus sophistiqués permettent d'ajouter des photos ou des dessins, ainsi que modifier la taille et l'apparence des caractères et vérifier l'orthographe. Les documents ainsi créés peuvent ensuite être imprimés autant de fois que nécessaire.[1]

1.1.1 Types des logiciels de traitement de texte

Un éditeur de texte est un logiciel qui permet de saisir et modifier interactivement des textes. En général, possède peu ou pas de fonctions de mise en forme. Des différents codages (par exemple en ASCII ou en Unicode) peuvent être utilisés pour convertir les signaux émis par les touches du clavier, en unités binaires représentant des caractères, qui sont ensuite affichés ou imprimés sous la forme de caractères d'imprimerie.[1]

1.1.2 Traitement de texte interactif (WYSIWYG)

Un traitement de texte interactif permet de contrôler, de manière interactive à travers une Interface Homme machine, la mise en forme d'un ensemble de textes (pouvant contenir des images) en un document (par exemple un CV, un rapport de thèse, un courrier, un périodique). Les traitements de texte modernes utilisent généralement les alphabets et autres caractères couverts par Unicode. Un logiciel WYSIWYG est un logiciel qui dispose d'une interface graphique qui permet à l'utilisateur de voir son document tel qu'il sera publié. Ces logiciels donnent de plus un accès intuitif aux fonctionnalités et permettent de cette manière de réaliser la mise en forme du document sans avoir à mémoriser et à utiliser des commandes complexes.[1]

Logiciel	Gratuit	Libre	Systèmes d'exploitation
Microsoft Word	Non	Non	Windows, Mac Os, Inclus dans Microsoft office
Quick Office (en)	Oui	Non	Androïde, IOS
Libre Office Writter	Oui	Non	Linux, Windows, Mac OSx, Inclus dans Microsoft office
Word Pad	Non	Non	Microsoft Windows, Inclus dans Microsoft Windows

Tableau 1.1: Exemple des logiciels traitement de texte WYSIWYG

1.1.3 Logiciels de traitements de texte (WYSIWYM)

Un traitement de texte automatique est un logiciel permettant de traiter automatiquement du texte ; en d'autres termes, c'est un compilateur pour un langage de mise en forme de textes : l'utilisateur utilise un éditeur de texte standard et compose un document (source) en utilisant un balisage sémantique, et dans un deuxième temps, le compilateur produit le document sous sa forme imprimable, ou consommable ,le compilateur combine le balisage sémantique (par exemple : descriptions en sections, blocs ou encore citation) avec une feuille de style pour produire la forme consommable du document. Cette forme est préférée pour tout ce qui concerne l'édition technique, et, a fortiori, lorsque l interviennent des chaînes complètes de sous-traitants, afin de garantir une homogénéité de présentation à chaque niveau, en fonction des choix de chacun. Un exemple de ce type de logiciel est TEX, dont le point fort est notamment la mise en page de documents scientifiques avec des formules mathématiques, mais qui permet, grâce à ses nombreux (paquets), de mettre en page un grand nombre de documents variés.
Le modèle du WYSIWYM s'oppose à celui du WYSIWYG qui lui est postérieur. Les logiciels basés sur ce modèle servent principalement à l'édition de documents textuels. Ils se caractérisent par le fait que l'auteur n'écrit pas le contenu qui sera lu par le lecteur. En effet, le lecteur n'a à sa disposition que la traduction du code sur l'interface d'un éditeur WYSIWYM (les balises du code et les possibilités de structuration du document lui sont proposées via cette interface). L'auteur n'écrit ainsi que du code à l'aide de l'éditeur. Lors de la phase de rédaction, il ne voit pas le résultat de sa publication telle que le lecteur pourra la voir. L'auteur est ainsi incité à se focaliser sur le sens du contenu et à

faire abstraction de son apparence finale, la forme étant gérée de manière automatique par le logiciel.
Exemple des logiciels traitement de texte WYSIWYM: [2]

- Lyx compatible LaTeX.
- Scientific Word et workplace compatible LaTeX.

1.2 Présentation de TeX:

TeX ce n'est pas à proprement parler un "formateur de texte", mais plutôt un "formateur de document", au sens ou il analyse un texte source pour fournir le résultat "compilé" sous forme d'un document final.[3]

1.2.1 Historique de TeX:

1978 : le mathématicien Donald Ervin. Knuth (crée TeX, avec son langage, son moteur de compilation TeX, et un ensemble de macros, PLAIN TeX, regroupées sous forme d'un format. Premier moteur tex sur 7 bits en entrée : \'e pour encoder é.
Rappel : en **1978**, peu de "Personnal Computer", écrans textuels, mémoires de quelques Ko...
Principe

- on part d'un fichier source en texte brut (.tex)
- "compilation" à l'aide du moteur tex et des macros de PLAIN TeX,
- fichier de description de la page en DVI (.dvi, "device indépendant").
- impression (drivers divers : dvips pour une sortie en POSTSCRIPT par exemple).

1982 : Lamport introduit LaTeX , un autre jeu de macros au-dessus de TeX, regroupées sous forme d'un format, plus simple à utiliser que PLAIN TeX. C'est surtout un langage de description sémantique du texte. La compilation utilise toujours le moteur tex. Apparition des packages (modules) en français) extension facilitée des fonctionnalités.

1989 : La version 3 du moteur tex permet de gérer des caractères sur 8 bits (256 caractères différents), donc de lire des textes avec des lettres accentuées

1994 : LaTeX2e remplace LaTeX2.09 qui vieillissait mal, en particulier par l'anarchie dans les noms des packages et leur incompatibilité.

Fin des années 1990 : Hàn Thê Thành introduit le moteur **pdftex** (son travail de thèse) : sortie par défaut dans le format PDF (portable document format inventé par Adobe), gestion des polices vectorielles, extensions micro-typographiques, accès à des fonctionnalités PDF (hyperliens, table des matières...). Aujourd'hui c'est le moteur par défaut dans les installations de TEX.

2008 : Le moteur tex atteint la version 3.1415926 (corrections de bugs).

- Aucune fonctionnalité n'est ajoutée à tex depuis la version 3.
- Chaque correction de bug ajoute une décimale de

$$\pi^2$$

- Première version publique du moteur xetex.
- Extension de pdftex pour utiliser les polices installées sur le système d'exploitation, codage en entrée UNICODE (16 bits).

2010 : Première version publique du moteur luatex. Fusion du meilleur de pdftex et de xetex, ouverture de la composition des pages au langage de programmation LUA. Futur proche (5 ans) : xetex et luatex vont remplacer pdftex : utilisation des dernières technologies en matière de polices vectorielles (TRUETYPE, OPENTYPE), en particulier le "standard" développé par Microsoft et Adobe sur les polices mathématiques. Futur toujours trop loin : Le projet LATEX3 doit remplacer LATEX2e. 20 ans qu'on attend..[4]

1.3 Présentation de LATEX

LATEX est une surcouche de TEXincluant un moteur et des macros ainsi que certaines notions assez pratiques, comme la notion d'environnements, assez proche des éléments SGML (et donc de HTML par exemple). Par rapport à TEX, LATEXpermet également le support d'arguments optionnels dans les macros, la notion de modules (packages) qui permettent de l'étendre. Du coup, la communauté d'utilisateurs a créé de nombreux packages, notamment en :

- Sciences : chimie, physique, mathématiques...
- Loisirs : musique...
- Langues : arabe, hébreu, langues asiatiques... [4]

1.3.1 Principe de LATEX

LATEX exige du rédacteur de (ou au moins pousse le rédacteur à) se concentrer sur la structure logique de son document, son contenu, tandis que la mise en page du document

(césure des mots, alinéas) est laissée au logiciel lors d'une compilation ultérieure. LaTeX sépare en deux phases la forme du contenu. Avec les logiciels de type WYSIWYG tels que LiberOfficeWriter ou Microsoft Word. La structure est codée par les styles, la forme étant automatiquement et immédiatement visible à l'écran. Des logiciels permettant de rédiger des documents LaTeX de cette façon existent (TexMacs..).
Il existe également d'autres méthodes, comme celle adoptée par le logiciel LyX, selon le concept de WYSIWYM, qui permet d'écrire un texte à l'écran sans avoir le rendu exact à l'écran lors de l'écriture, mais dont le résultat visuel à l'exportation choisie (DVI, PostScript ou bien PDF) aura toutes les qualités du rendu d'un document LaTeX compilé, puisque LaTeX est utilisé en arrière-plan. Des langages de balisage léger tels que txt2tags, reStructuredText, ou Plain Old documentation permettent également d'exporter vers LaTeX, au prix d'une certaine limitation dans l'accès aux fonctionnalités avancées de LaTeX.[4]

1.3.2 Utilisation de LaTeX

Apprendre à produire des documents avec LaTeX cela demande forcément un peu d'investissement, même s'il est possible d'apprendre au fur et à mesure de ses besoins. Cela vient du fait que c'est un système complètement diffèrent des traitements de texte que vous avez l'habitude d'utiliser. Par exemple, pour écrire diffèrent en italique, je n'ai pas cliqué sur un bouton dans un menu comme je le ferais avec Word, mais en gros, j'ai saisi dans mon fichier quelque chose comme : **DébutItalique** différent **FinItalique**. Tout ce texte est construit selon ce principe, ensuite LaTeX parcourt ce fichier et le transforme en un beau document, selon ces indications.

1.3.2.1 Création des articles Scientifiques

On peut écrire facilement des formules de maths, dessiner des arbres, des molécules, des diagrammes commutatifs, etc... Dans chaque communauté de scientifiques, certains ont créé des packages (fichiers d'options), on peut télécharger gratuitement ces packages à travers l'Internet, pour adapter LaTeX à leurs besoins.

1.3.2.2 La qualité typographique

C'est difficile à comprendre, mais les traitements de texte classiques n'ont pas été conçus avec l'expertise de typographes, contrairement au système LaTeX. On en est assez vite convaincu à la vue d'un document LaTeX: espace entre les caractères, césures, arrangement des paragraphes, mais également disposition des figures dans le texte, domaine pour lequel Word est très mal conçu. D'autre part, LaTeX fournit beaucoup moins de polices, cela vient du fait que le document produit doit être mondial, peut être visionné sur n'importe quelle version de n'importe quel système d'exploitation.

1.3.2.3 Pour créer des gros documents

C'est pour cette qualité de LaTeX que l'investissement est le plus rentable. Pour cela, nous le choisissons pour la gestion de toutes les choses compliquées liées à la production de gros documents comme (les Livres, Rapport de recherche, Mémoire et Thèse..) :
Exemple de technique LaTeX pour ces Gestion :

1. Numérotation automatiquement les sections, sous-sections, appendices, figures, formules, théorèmes, etc...
2. LaTeX prend en charge tous seule la création de la table des matières et l'index.
3. Numérotation très facilement les équations, les formules, les tableaux puis faire référence à ces numéros.
4. Il gère très bien la disposition des figures dans un texte.
5. On peut se rejoindre très facilement plusieurs documents, Permettre à un groupe de personnes de travailler sur différents chapitres du même document.
6. La Taille de document LaTeX ne prennent que très peu de place sur le disque, en contraire aux documents produits par un traitement de texte

1.3.2.4 Pour la pérennité

C'est un critère déterminant pour une thèse. Rien ne permet de dire qu'un document écrit en Word aujourd'hui puisse être parfaitement lisible (et modifiable) dans 10 ou 15 ans. Au gré des versions, des options disparaissent ou sont créées. Ce n'est pas le cas avec LaTeX . Les modifications qui y sont apportées ne se font jamais au détriment des utilisateurs, et pour cause, c'est un système entièrement gratuit.

1.3.2.5 Pour la souplesse:

Le principe même de LaTeX , c'est un noyau commun, qui permet de créer tous les documents simples, et la possibilité de créer des nouveaux modules adaptés à des besoins particuliers. Concevoir ces modules demande beaucoup de talent en programmation mais dans chaque communauté ont été développées des bibliothèques spécifiques qui sont disponibles sur Internet : mathématiques, informatique, chimie, mais aussi partitions de musique, parties d'échecs, russe, grec, etc. On peut donc utiliser simplement LaTeX en se servant des modules créés par d'autres utilisateurs sans les concevoir soi-même.
En revanche, il est très simple de créer des petites commandes adaptées à ses besoins. Par exemple, si j'en ai assez de saisir au clavier " Organisation des Nations Unies ", je peux créer une commande **ONU** . A chaque fois que LaTeX va lire cette commande, il va automatiquement la traduire en "Organisation des Nations Unies".

1.3.2.6 Parce que c'est universel

Pour échanger des documents produits avec LaTeX, on peut les transformer en fichier PostScript (lisible par toutes les imprimantes), PDF, ou même HTML. Peu importe que la personne avec qui je travaille utilise un PC avec Windows, Linux ou un Mac. Et bien s ûr, tout ¸ca est entièrement gratuit.

1.3.3 Installation de LaTeX

Pour utiliser LaTeX, Il faut installer une distribution correspondant au système d'exploitation. Les distributions fournissent des programmes permettant d'automatiser la configuration et l'installation de LaTeX, TeXet tous les utilitaires connexes :[5]

Système Exploitation	**Configuration & installation**
Unix	on trouve encore la distribution TeTeX. la développement de ce distribution ait été stoppé en 2006. Aujourd'hui on installe généralement la TeXLive
MacOS	la distribution de référence est MacTeX
Windows	le plus simple est sans doute de choisir proTeXt qui installe la distribution MiKTEX et quelques outils de développement dont un programme de visualisation de fichiers au format PostScript (gsview)

Tableau 1.2: Configuration et installation Latex

Il faut parfois ajouter à ces distributions (si elles n'en contiennent pas déjà un) un éditeur de texte puisque vous le découvrirez bien assez tôt, utiliser LaTeX c'est taper du texte et des commandes dans des fichiers. La production d'un document avec LaTeX consiste à traduire (on dit aussi compiler) une source " donc créé par un éditeur de texte " en un format destiné à l'affichage ou à l'impression. Il existe donc, plus ou moins intégrés aux distributions, des outils célèbres pour la visualisation des différents fichiers résultants de la compilation

Format DVI xdvi, kdvi sous Unix et yap sous Windows font partie des programmes permettant de visualiser le résultat de la compilation d'un fichier LaTeX.

Format PostScript la suite GhostScript disponible sous des noms qui peuvent varier selon la plateforme, permet de visualiser des fichiers au format PostScript.

Format PDF mis à part le célèbre Acrobat Reader, il existe sous Unix des utilitaires permettant de visualiser le format Pdf : xpdf, evince,ps ...

Autres fichiers LaTeX:

Fichier extension .tex Ce sont les fichiers contenant toutes les commandes que vous allez taper, i.e les fichiers sources.

Fichier extension .bib et .bbl Ces fichiers servent à la gestion de la bibliographie.

Fichier extension .bib et .bbl .aux, .toc, .idx Ces fichiers sont utilisés par LaTeX pour gérer les références dans votre document

1.3.4 Cycle de production

Même si LaTeX n'est pas à proprement parler un langage de programmation compilé, on peut malgré tout faire une analogie entre le cycle de production d'un document LaTeX et le cycle édition-compilation-exécution d'un développement de logiciel avec un langage de programmation classique.

1.3.4.1 Édition

Un document source LaTeX est un fichier texte. Ainsi la manipulation d'un fichier LaTeX ne demande pas de logiciel particulier, si ce n'est un éditeur de texte classique.

1.3.4.2 Compilation

La compilation génère un jour ou l'autre des erreurs.. En tout cas, après suppression des erreurs de compilation, on obtient un fichier portant l'extension **dvi** pour device indépendant. Ce qui signifie que le fichier contient des informations indépendantes du périphérique de sortie (écran, imprimantes, ...). Ce fichier de type binaire contenant une "image" du document portable sur tout système Tex quel que soit le système d'exploitation. Il existe ensuite des programmes permettant soit :

- **de visualiser le document:** dvi ! bitmap écran.
- **de l'imprimer:** dvi ! langage imprimante.
- **de le convertir :** dvi ! fichier PostScript.

1.3.4.3 Visualisation

La visualisation s'effectue simplement -après compilation sans erreur - grâce au programme xdvi.

1.3.4.4 Impression

Le fichier **dvi** peut également être imprimé. Là également il y a différents programmes qui existent en fonction de la plate-forme et même parfois de l'imprimante.

1.3.4.5 Conversion en PostScript

Le fichier **dvi** nécessite de nombreuses polices de caractères différentes pour être visualisé ou imprimé. Si on veut le lire sur un autre ordinateur, il faut impérativement avoir toutes ces polices, ce qui n'est pas toujours le cas. Elles peuvent être générées automatiquement mais il faut du temps et de l'espace disque.

1.3.5 Le document LaTeX

Le document LATEX est créé dans un fichier **.tex** qui est rédigé obligatoirement en deux parties :

1.3.5.1 Le préambule

c'est le partie de "déclaration", dans cette partie, on déclare :

- Le type de document à créer (article, livre, lettre, présentation ...)
  Par exemple :**\documentclass{ article}** dans le préambule signifie que l'on rédige un texte court, autant que **\documentclass{ report}** dans le préambule signifie que l'on rédige un texte moyen ou long.
- Les packages nécessaires (chaque document exige un environnement particulier de rédaction appelé package comprenant la langue et le type d'encodage à utiliser, par exemple). Il s'agit en fait d'extensions auxquelles on fait appel ou non en fonction des documents à réaliser.
  Par exemple : **\usepackage{graphicx}** dans le préambule signifie que l'on rédige un texte où l'on veut insérer des graphiques

1.3.5.2 Le corps du texte

Cette partie débute avec la commande:**\begin{document}** et se termine par :**\end{document}** . Là, le texte du document est rédigé et accompagné des commandes de mise en forme souhaitées.

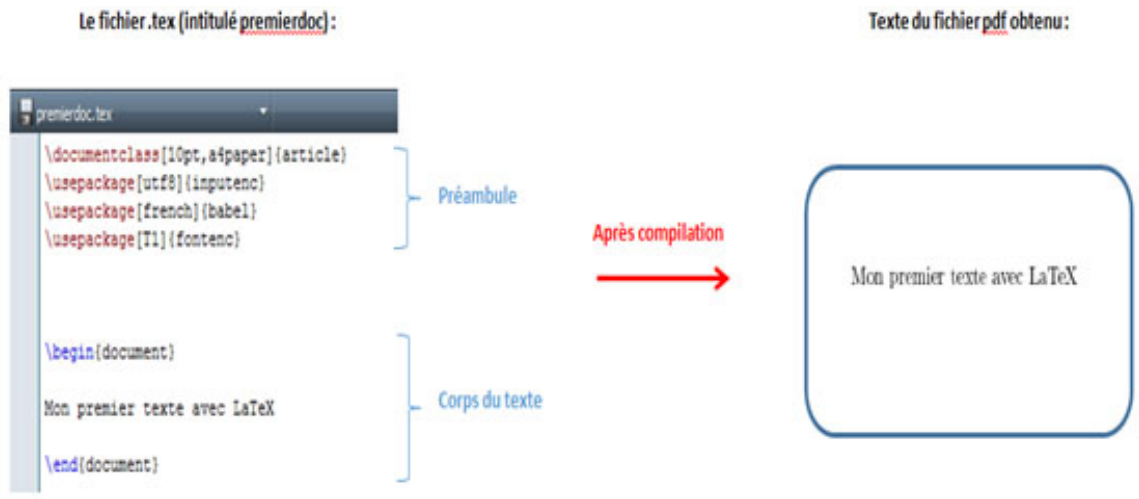

Figure 1.1: Exemple d'un Fichier Simple

1.4 Les tableaux sous LaTeX

Cette section décrit comment insérer des tableaux de données dans un document. On va commencer par voir les environnements **tabbing** et **tabular** qui sont les plus utilisés. On verra ensuite comment ajouter une légende à un tableau en utilisant l'environnement table. Après cela, on verra comment personnaliser l'aspect des tableaux :[6] [7]

1. Composer des tableaux simples:
    - Insertion d'un tableau.
    - Filets d'un tableau.
    - Legends d'un tableau.
    - Largeur des colonnes.
    - Largeur d'un tableau.
2. Quels Opérations sur les cellules:
    - Fusionner des cellules dans même ligne.
    - Fusionner des cellules dans même colonne.
    - Fusionner des cellules sur deux lignes et deux colonnes.
    - Insérer une barre oblique dans une cellule.
    - Rotation du texte.

1.4.1 Composer des tableaux simples

1.4.1.1 Insertion d'un tableau

Pour insérer un tableau on utilise l'environnement **tabular** qui prend un paramètre qui décrit les colonnes du tableau. Une nouvelle colonne est définie à l'aide d'une lettre qui décrit l'alignement horizontal du texte dans cette colonne. Les valeurs possibles sont :

l (left)	Appuie le texte de la colonne sur la gauche
r (right)	Appuie le texte de la colonne sur la droite
c (center)	Centre le texte dans la colonne
p{l}	Le texte sera traité comme un paragraphe dans une colonne de largeur l, et pourra être émis sur plusieurs lignes si besoin
| (pipe)	Trace un trait vertical pour séparer deux colonnes

Tableau 1.3: les valeurs d'alignement de texte

- Dans l'environnement **tabular** : la commande **hline** trace les filets horizontaux.
- L'esperluette **&** sert de séparateur des colonnes.
- Le symbole suivant \\ indique la fin de la ligne.

Exemple de création d'un Tableau:
le code source:
\begin{tabular}|c|c|l|r| \hline
Un tableau & quatre colonnes & avec des cellules centrées & d'autres alignées à droite ou à gauche \hline
centrée & centrée & à gauche & à droite \hline \hline
un & deux & trois & quatre \hline
\end{tabular}

1.4.1.2 Filets d'un tableau

On peut ajouter des filets verticaux et horizontaux à un tableau. Les filets verticaux sont définis dans l'option de l'environnement **tabular** . Il suffit d'insérer | partout là où on souhaite un filet vertical. Les filets horizontaux sont définis avec les données du tableau. La commande **hline** insère un filet horizontal. Elle doit toujours être placée avant les données de la ligne. Il est également possible d'avoir des filets horizontaux partiels qui ne s'étendent que sur certaines colonnes. On les définit à l'aide de la commande **cline** qui prend en paramètre les colonnes sur lesquelles le filet doit s'étendre. Pour ajouter des filets horizontaux du tableau:
le code source:
\begin{tabular}{cc} \hline
cellule 1 & cellule 2 \\ \hline
cellule 3 & cellule 4 \\ \hline
\end{tabular}
Résultat de PostScript:

cellule 1	cellule 2
cellule 3	cellule 4

Tableau 1.4: Résultat PostScript Fillets d'un tableau

On utilise l'environnement **center** pour centrer le tableau dans la page. On peut aussi obtenir un filet double horizontal en doublant la commande **hline** .
Exemple de centralise un tableau:

Code source:

```
\begin{center}
\begin{tabular} l | c || r |  \hline
1 & 2 & 3 \\  \hline
4 & 5 & 6 \\  \hline
7 & 8 & 9 \\  \hline}
\end{tabular}
\end{center}
```

Résultat en Postscript

1	2	3
4	5	6
7	8	9

Tableau 1.5: Résultat PostScript Centralisation d'un tableau

1.4.1.3 Légende d'un tableau

Pour ajouter une légende à un tableau, il faut le placer dans un environnement table et utiliser la commande **caption**. L'utilisation de l'environnement de table définit un nouvel objet flottant, donc nous n'avons plus de contrôle absolu dans la position du tableau. Afin de centrer le tableau, il faut utiliser la commande **centering**. Nous prouvons placer la légende au-dessus ou en-dessous du tableau en changeant la position de la commande**caption**.

Exemple de Légende d'un tableau:

le code source:

```
\begin{table}
\begin{center} \begin{tabular}|c|c|c|   \hline
Voici & un tableau & et une légende\\  \hline
\end{tabular}
\end{center}
\caption{Résulat postScript Légende tableau }
\end{table}
```

Résultat en PostScript:

Voici	un tableau	et une légende

Tableau 1.6: Résultat en PostScript Légende tableau

1.4.1.4 Largeur des colonnes

Les colonnes de type "c", "l" et "r" ne permettent pas de retour à la ligne. La largeur des colonnes est celle de la plus grande ligne de cette colonne. Ce qui peut très vite devenir inesthétique et provoquer des débordements. Pour résoudre ce problème, on va utiliser le descripteur de colonnes p<taille> où <taille> sera la longueur souhaitée de la colonne dans une unité reconnue par LaTeX.

Exemple de 'Largeur des colonnes':

Code source:

```
\begin{tabular}|p3cm|p3cm| \hline
Une cellule contenant un texte très long & Une autre cellule contenant un texte encore plus long, beaucoup, beaucoup plus long \\  \hline \end{tabular}
```

Résultat en PostScript:

Une cellule contenant un texte très long	Une autre cellule contenant un texte encore plus long, beaucoup, beaucoup plus long

Tableau 1.7: Résultat en PostScript largeur des colonnes

Les deux colonnes ont maintenant d'une largeur de 3cm, mais nous pouvons constater que ce résultat n'est pas du meilleur effet. Contrairement aux colonnes "c", "l", "r", les colonnes de type "p" acceptent les commandes telles que :

- \centering.
- \raggedright.
- \raggedleft.
- \par.

Les commandes telles que **\centering** et **\raggedright** redéfinissent
, il va donc falloir utiliser **\tabularnewline** à la place de \\.

Exemple utilise \raggedright et \centering :

```
\begin{tabular}{|p3cm|p{3cm}|} \hline
\raggedright
Une cellule contenant un texte très long & Une autre cellule contenant un texte encore plus long, beaucoup, beaucoup plus long \raggedright \tabularnewline \hline
\end{tabular}
\hfill
```

\begin{tabular}{|p{3cm}|p{3cm}|}
\hline \centering
Une cellule contenant un texte très long & \centering
Une autre cellule contenant un texte encore plus long, beaucoup, beaucoup plus long \tabularnewline
\hline
\end{tabular}

Résultat de PostScript:

Une cellule contenant un texte très long	Une autre cellule contenant un texte encore plus long, beaucoup, beaucoup plus long
Une cellule contenant un texte très long	Une autre cellule contenant un texte encore plus long, beaucoup, beaucoup plus long

Tableau 1.8: Exemple utilise \raggedright et \centering

1.4.2 Quels Opération sur les cellules

1.4.2.1 Fusionner des cellules dans même ligne

La commande \multicolumn{<nombre de cellules à fusionner>}{<descripteur de colonne>}{<texte>} permet de fusionner les colonnes d'une même ligne.

Exemple de Fusionnement:

Code source:

\begin{tabular}{|c|c|c|} \hline
1 & 2 & 3 \\ \hline
b & \multicolumn{2}{c|}{a}\\ \hline
\end{tabular}

Résultat en PostScript:

1	2	3
b	a	

Tableau 1.9: Fusionner des cellules dans même ligne

1.4.2.2 Fusionner des cellules dans même colonne

Pour fusionner les cellules d'une même colonne, on utilisera l'extension **multirow**. Il faut donc ajouter \usepackagemultirow dans le préambule et utiliser dans le tableau la commande \multirow{<nombre de lignes à fusionner>}{<taille>}{<texte>}. Où **<taille>** est la largeur de la cellule, elle peut être remplacée par ***** qui ajustera automatiquement la taille au contenu de la cellule.

Exemple de fusionnement des cellules
Code source:

```
\begin{tabular}{|c|c|c|} \hline
1 & 2 & 3\\  \hline
\multirow{2}{*}{A}&4&5\\  \cline
{2-3} & 6 &7 \\  \hline
\end{tabular}
```

Résultat en PostScript:

1	2	3
A	4	5
	6	7

Tableau 1.10: Fusionner des cellules dans même colonne

1.4.2.3 Fusionner des cellules sur deux lignes et deux colonnes

Il est possible de combiner les commandes **multirow** et **multicolumn** pour fusionner plusieurs cellules juxtaposées.

Exemple de fusionnement des cellules sur deux lignes et deux colonnes Code source:

```
\begin{tabular}{|c|c|c|} \hline
1 & 2 & 3 \\\hline
\multicolumn{2}{|c|}{\multirow{2}{*}{ABC}} & 4 \\  \cline{3-3}
\multicolumn{2}{|c|}{} & 5 \\  \hline
\end{tabular}
```

1	2	3
ABC		4
		5

Tableau 1.11: Fusionner des cellules sur deux lignes et deux colonnes

1.4.2.4 Insérer une barre oblique dans une cellule

La commande **backslashbox** du package **slashbox**, nous pouvons diviser une cellule en deux parties séparées par une barre oblique. Autant que, la commande **slashbox** permet d'avoir une cellule divisée par rapport à l'autre diagonale.

Exemple d'insertion d'un barre oblique ::

Code source:

```
\begin{tabular}[b]{|l|c|c|c|} \hline
\backslashbox{Produit}{Année} & 1999 & 2000 & 2001 \\
\hline \hline
Livre & 15 & 10 & 7 \\
CD & 10 & 17 & 22 \\
\hline
\end{tabular}
```

Résultat en PostScript:

Produit \ Année	1999	2000	2001
Livre	15 10		7
CD	10	17	22

Tableau 1.12: Insertion une barre oblique dans une cellule

1.4.2.5 Rotation du texte

On souhaite parfois effectuer une rotation du texte dans une cellule. Il suffit d'utiliser la commande **rotatebox** du package **graphicx**

Exemple:

Code source:

```
\begin{tabular}{ |c|c|c| } \hline \multirow{8}{*}[-0.4ex]{\rotatebox{90}{Some text}}
& row 1 & row 1 \\
& row 2 & row 2 \\
& row 3 & row 3 \\
& row 4 & row 4 \\
& row 5 & row 5 \\
& row 6 & row 6 \\
& row 7 & row 7 \\
& row 8 & row 8 \\
\hline \end{tabular}
```

Résultat en PostScript:

Some text	row 1	row 1
	row 2	row 2
	row 3	row 3
	row 4	row 4
	row 5	row 5
	row 6	row 6
	row 7	row 7
	row 8	row 8

Tableau 1.13: Rotation Texte dans une cellule

Conclusion

Dans ce chapitre nous avons vu quelques notions de base sur LaTeX telles que notions de documents, de classes, . . . etc. En plus, nous avons vu les différentes commandes qui permettent de créer un tableaux. Nous dois admettre que la syntaxe d'un tableau en LaTeX peut sembler plutôt tordue. Mais heureusement, nous en avons assez vu que vous puissiez créer n'importe quel tableau selon le besoin dans le document.

Dans le prochain chapitre, nous allons présenter la conception d'un système qui permet de générer des tableaux simples en LaTeX.

Chapitre 2

Analyse et Conception

Introduction

L'approche objet est incontournable dans le cadre du développement de systèmes logiciels complexes, capables de suivre les évolutions incessantes des technologies et des besoins applicatifs. Ce chapitre est consacrée aux étapes fondamentales pour le développement de notre outil de design des tableaux pour LaTeX pour la conception de notre application, nous avons utilisé un langage de modélisation très complet, qui couvre de nombreux aspects du développement des logiciels, comme les exigences, l'architecture, les structures et les comportements, c'est-a-dire on travaille avec le formalisme UML (sigle désignant l'Unified Modeling Language) qui permet de modéliser un problème de façon standard.

Dans ce chapitre, nous présentons quelques définitions et concepts du langage UML. Ensuite, nous révélons la modélisation de notre système que nous utilisons trois cartes les plus célèbres dans ce type d'application, ces cartes sont les suivantes [8]:

- Diagramme de cas d'utilisation.
- Diagramme des séquences.
- Diagramme de classe.

2.1 Présentation de l'UML

Dans cette section, nous allons voir une petite description du langage de modélisation UML.

2.1.1 Hisorique

UML est un langage de modélisation objet. Il est né de la fusion des trois méthodes qui ont le plus influencé la modélisation objet au milieu des années 90 : OMT, Booch et OOSE. Issu "du terrain" et fruit d'un travail d'experts reconnus, UML est le résultat

d'un large consensus. De très nombreux acteurs industriels de renom ont adopté UML et participent à son développement. La figure suivante nous donne un résumé sur le développement de UML:[9]

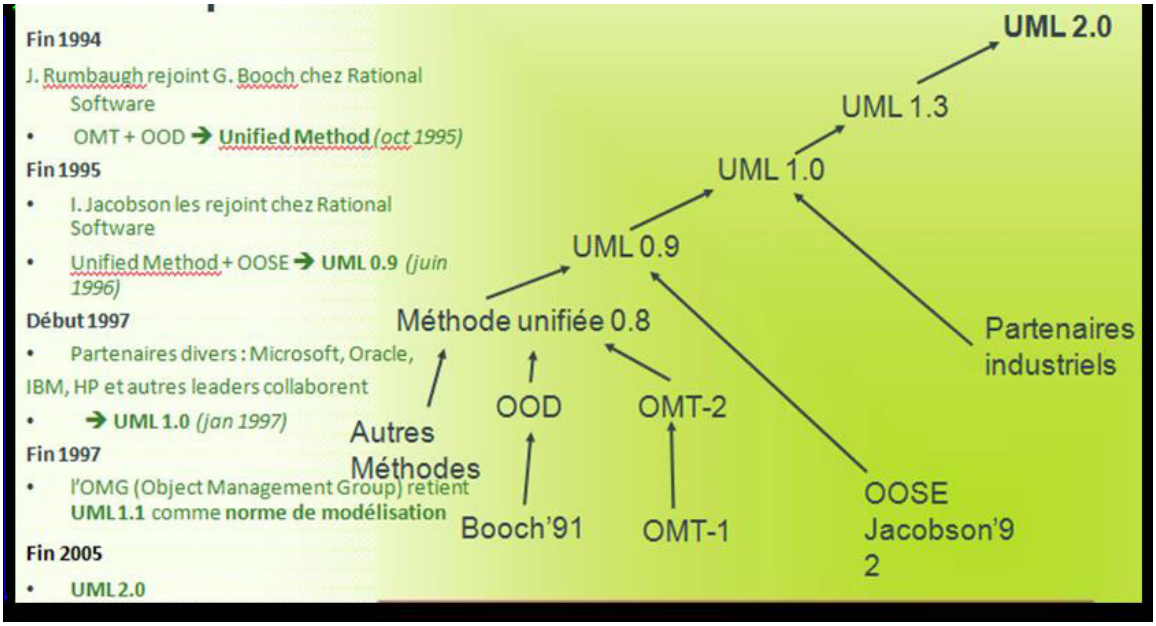

Figure 2.1: Schéma représentatif de l'évolution d'UML

2.1.1.1 Définition

UML c'est l'acronyme anglais pour (Unified Modeling Language) . On le traduit par (Langage de modélisation unifié). La notation UML est un langage visuel constitué d'un ensemble de schémas, appelés des diagrammes, qui donnent chacun une vision différente du projet à traiter. UML nous fournit donc des diagrammes pour représenter le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel.[8]

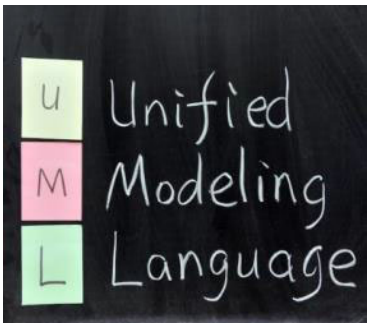

Figure 2.2: Logo d'UML

2.1.1.2 Les Diagrammes de UML

Il existe treize types de diagrammes sous UML qui sont dépendants hiérarchiquement. Ces diagrammes se complètent pour modéliser un système, et sont regroupés dans deux

grands ensembles :

- Les diagrammes statiques (ou structurel).
- Les diagrammes dynamique (ou comportemental).

Donc, on peut résumer les diagrammes de UML dans la figure suivante :

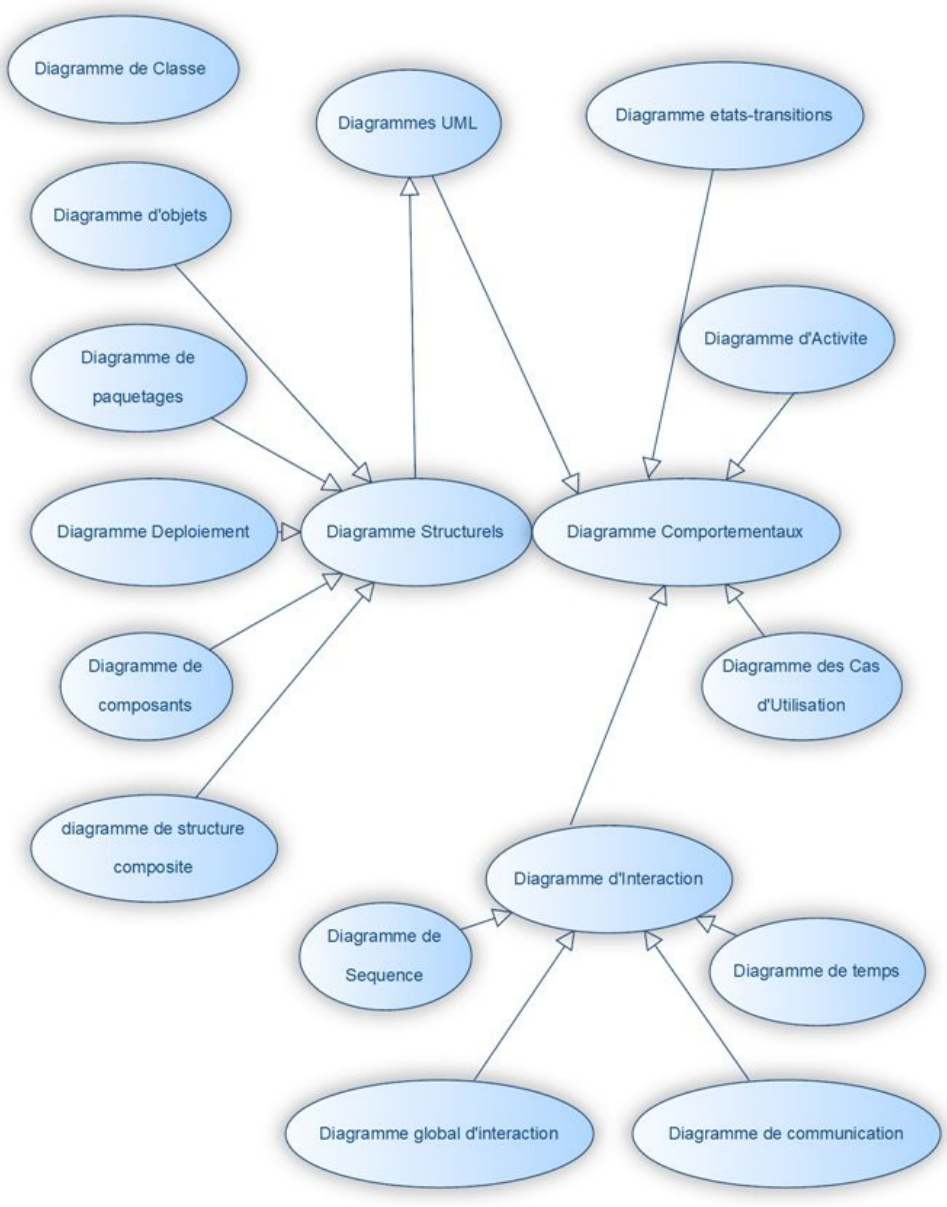

Figure 2.3: Les Diagrammes de UML

2.2 Diagramme de Cas d'utilisation

Ce diagramme représente la manipulation des systèmes ou des classes tels que n'importe qui peut le voir. Il divise les fonctions du système en unités cohérentes. Les cas d'utilisation sont utilisés pour exprimer les besoins des utilisateurs du système, telle que la vision orientée vers l'utilisateur est nécessaire au lieu de voir l'ordinateur.

2.2.1 Identification des Acteurs

Dans notre système, nous avons un seul acteur, c'est l'utilisateur d'application qui représente l'utilisateur du système. Il peut faire les tâches suivantes:

1. Assistant un tableau.

2. Fusionner des cellules.
3. Fractionner une cellule.
4. Ajouter une ligne.
5. Supprimer une ligne
6. Ajouter une colonne.
7. Supprimer une colonne.
8. Ajouter un texte.
9. Ajouter un diagonale
10. Générer le code LATEX.
11. Copier le code LATEX produit.

Le diagramme suivant détermine les différents cas d'utilisation:

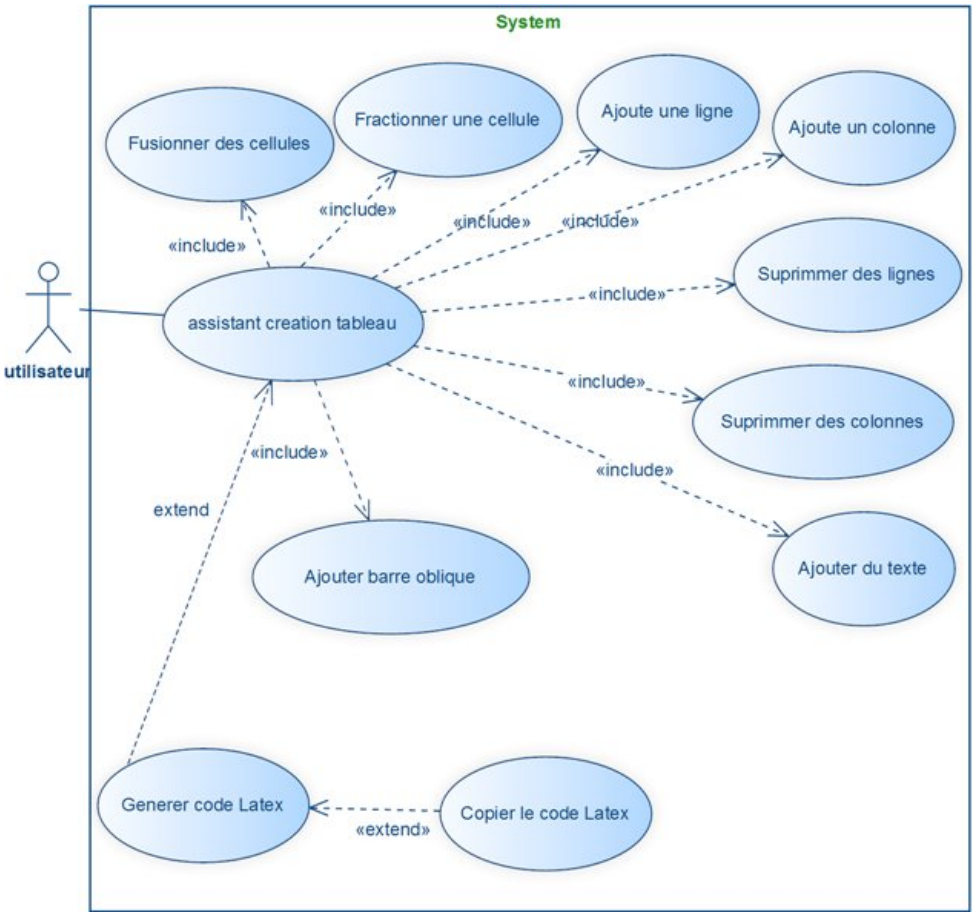

Figure 2.4: Diagramme de cas d'utilisation du système

2.3 Diagramme de Séquence

Les diagrammes de séquence présentent la coopération entre différents objets. Les objets sont définis et leur coopération est représentée par une séquence de messages entre eux.

Les objets peuvent être connectés à des classes existantes ou bien être créés indépendamment de toute classe. Si les objets sont connectés à des classes, les messages peuvent être connectés à des opérations.

Les diagrammes de séquence sont créés dans les interactions.
Dans ce qui suite, nous présentons les interactions des l'utilisateur avec le système à travers un diagramme de séquence.

Diagramme de séquence "Assistante Tableau"

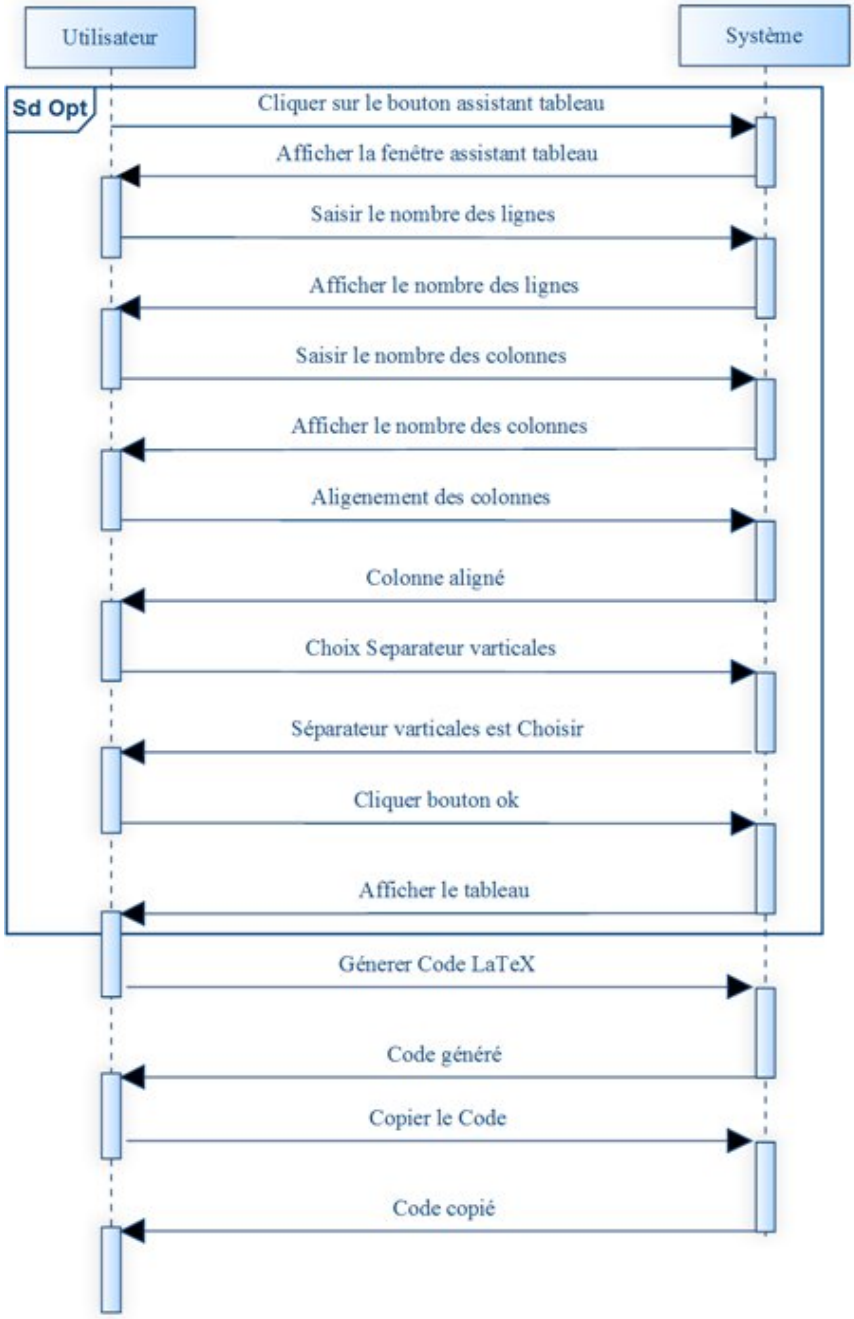

Figure 2.5: Diagramme de Séquence d'assistant du Tableau

Diagramme de Séquence "Fusionner des cellules"

Figure 2.6: Diagramme de séquence fusionnement des cellules

Diagramme de Séquence "fractionner une cellule"

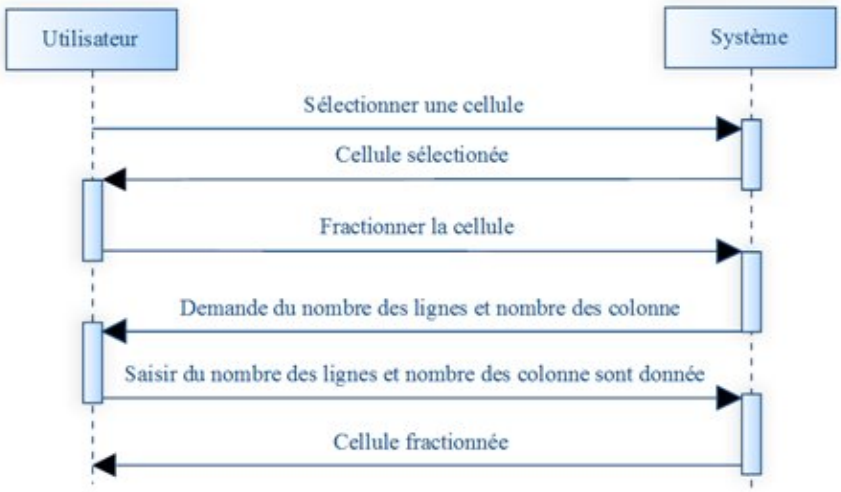

Figure 2.7: Diagramme de Séquence fractionner une cellule

Diagramme de Séquence "Ajouter un ligne"

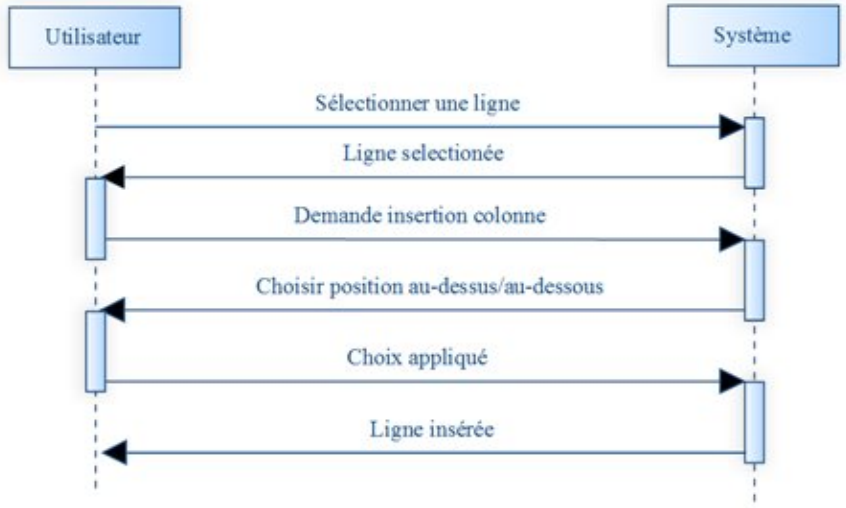

Figure 2.8: Diagramme de séquence Ajouter un ligne

Diagramme de Séquence supprimer un ligne

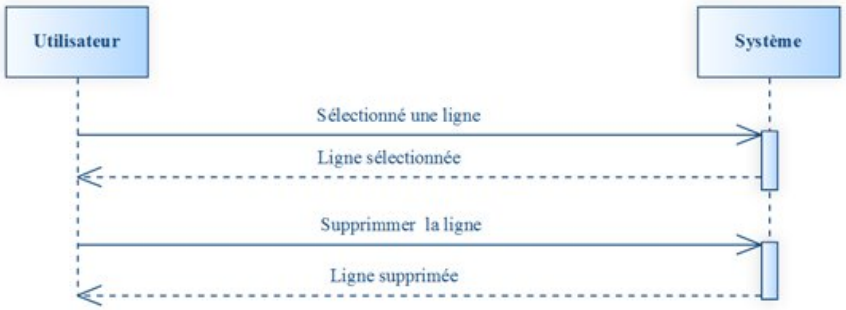

Figure 2.9: Diagramme de séquence suppression d'un ligne

Diagramme de Séquence "ajouter une colonne"

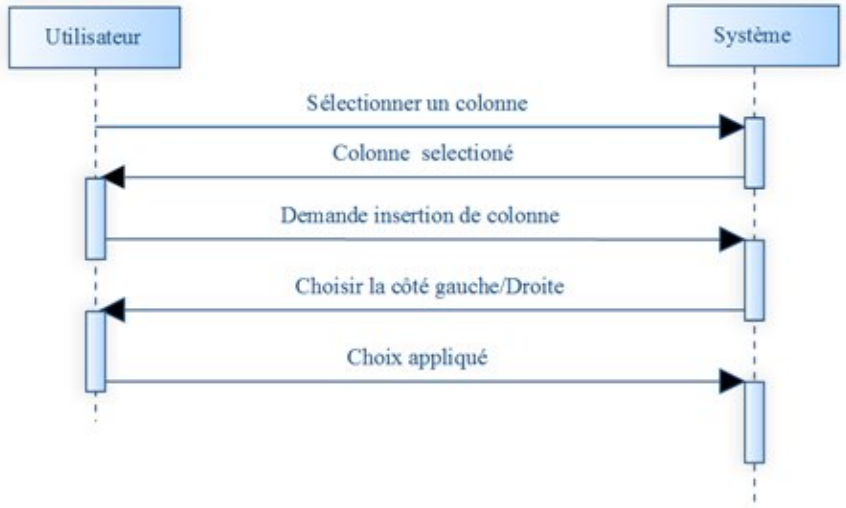

Figure 2.10: Diagramme de séquence d'insertion une colonne

Diagramme de Séquence "supprimer une colonne"

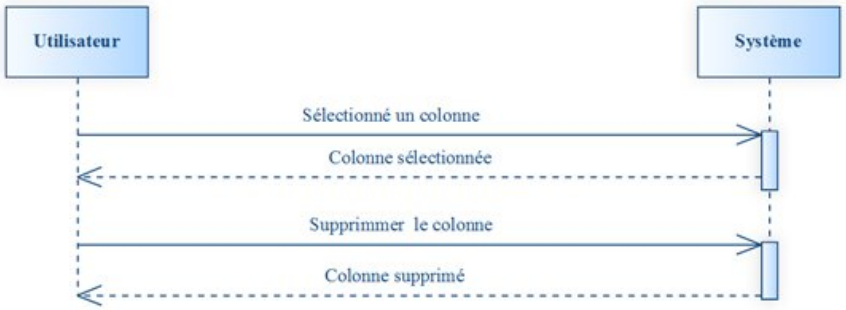

Figure 2.11: Diagramme de séquence de suppression d'un colonne

Diagramme de Séquence "Insertion d'un Texte"

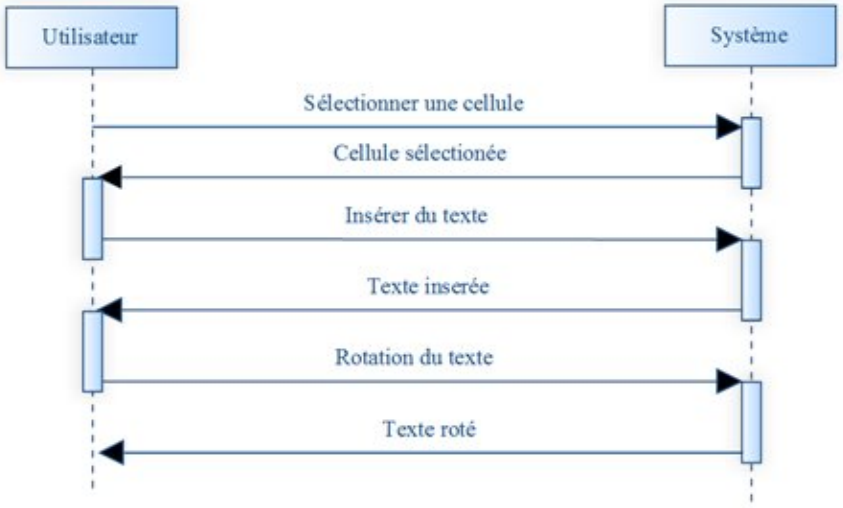

Figure 2.12: diagramme d'insertion d'un texte

Diagramme de séquence "Insertion d'un ligne diagonale"

Figure 2.13: Diagramme de séquence d'insertion d'un ligne diagonale

2.4 Diagramme de Classe

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation.

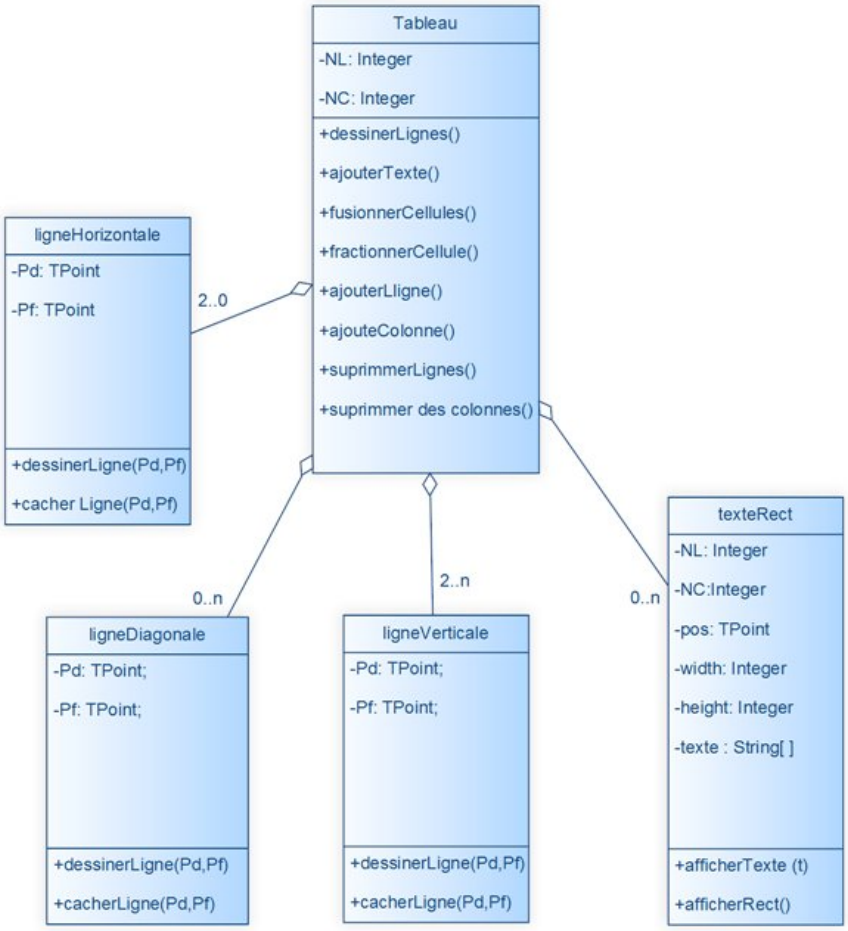

Figure 2.14: Diagramme de Classe

Conclusion

Dans ce chapitre, nous avons effectué la modélisation de notre système de design des tableaux pour Latex. La conception a été faite en utilisant le langage de modélisation UML. Les différents diagrammes de cas d'utilisation (Use Case), de séquences et de classes ont été présentés pour comprendre le déroulement de l'application. Une fois la conception faite, nous arrivons à la phase de développement et de réalisation de l'application qui doit s'appuyer sur la conception.

Chapitre 3

Réalisation:

Introduction

Le présent chapitre est consacré à la réalisation pratique proprement dite. Nous citons l'environnement de développement et les outils utilisés tels que les logiciel supplémentaires et le langage de programmation choisi. Enfin nous présenterons des illustrations des principales interfaces de notre application.

3.1 L'environnement de développement

Dans cette section nous allons voir une petite description de l'environnement de programmation Delphi ainsi que les caractéristiques de la version utilisée.

3.1.1 Langage de programmation (Delphi)

Nous avons développé notre application par le langage de programmation Delphi. Ce dernier est un langage informatique orienté objet qui est dérivé du langage Pascal et un environnement de développement pour plate-forme Microsoft. Il a été conçue par la société Borland dans les années 1980 puis revendu en 2007 à la société Enbarcadero. L'environnement de développement dispose de nombreux outils notamment une palette de composants très riche permettant de définir les interfaces graphiques Windows, WEB et bases de données Dans notre travail, nous avons utilisé la version RAD Studio 10.2.2 Tokyo.[10]

3.1.2 Caractéristiques de RAD Studio 10.2.2

La version de l'environnement Delphi utilisée dans ce travail possède les caractéristiques suivants:

- Possède un thème Dark conçu pour le travail de nuit ou une utilisation prolongée, avec des icônes de composants mises à jour, qui prennent en charge à la fois les thèmes clair et sombre.

- une amélioration de la productivité grâce à une fonction d'édition rapide de FireMonkey.
- quatre contrôles graphiques VCL Windows, conçus avec Windows 10 (TCardPanel, TStackPanel, TDatePicker et TTimePicker).
- une page d'accueil de l'EDI mise à jour avec des exemples de projets, des vidéos de la chaîne d'Embarcadero sur YouTube et un calendrier d'événements locaux et mondiaux pour aider les nouveaux utilisateurs à démarrer rapidement.

3.2 Matériel utilisé pour réaliser l'application

Afin d'accomplir ce travail. Nous avons utilisé une machine avec les caractéristiques suivantes:

- Laptop TOSHIBA.
- Processeur : Intel(R)Core(TM) i3-2310U CPU @ 2.10 GHZ
- Ram 4.00 GO.
- Disque dur : 650 GO.
- Système d'exploitation : Windows 8.1 Entreprise - 32bit

3.3 Logiciels supplémentaire utilisés

Durant ce travail, nous avons utilisé quelques logiciels et programmes supplémentaires, à savoir:

- Star UML 2.0.
- TeXstudio 2.12.2.
- PhotoShop CS6 Portable version 13.0 x32

3.4 Description de l'application

Dans cette section, nous allons voir une description des différentes interfaces importantes de l'application.

3.4.1 Interface Principale

Dans notre application il y a une seul interface principale, qui apparaît une fois le programme exécuté, l'utilisateur peut utiliser les fonctions du programme directement.

3.4.1.1 Barre de menu Application

Le menu principal de l'application contient les items suivants:

1. Fichier:
    - Assistant création Tableaux.
    - Ouvrir Fichier.
    - Enregistrer Sous.
    - Quitter l'application
2. Édition:
    - Ajouter Ligne.
    - Suppression Ligne.
    - Ajouter Colonne.
    - Suppression Colonne
    - Ajouter Texte.
    - Ajouter Ligne diagonale
    - Fusionnement.
    - Fractionnement.
3. Outils:
    - Génération Code LaTeX
    - Voir ongle Tableau
    - Voir Ongle Code LaTeX
4. Aides:
    - Aides.
    - A propos.

La figure ci-dessous explique l'interface principale:

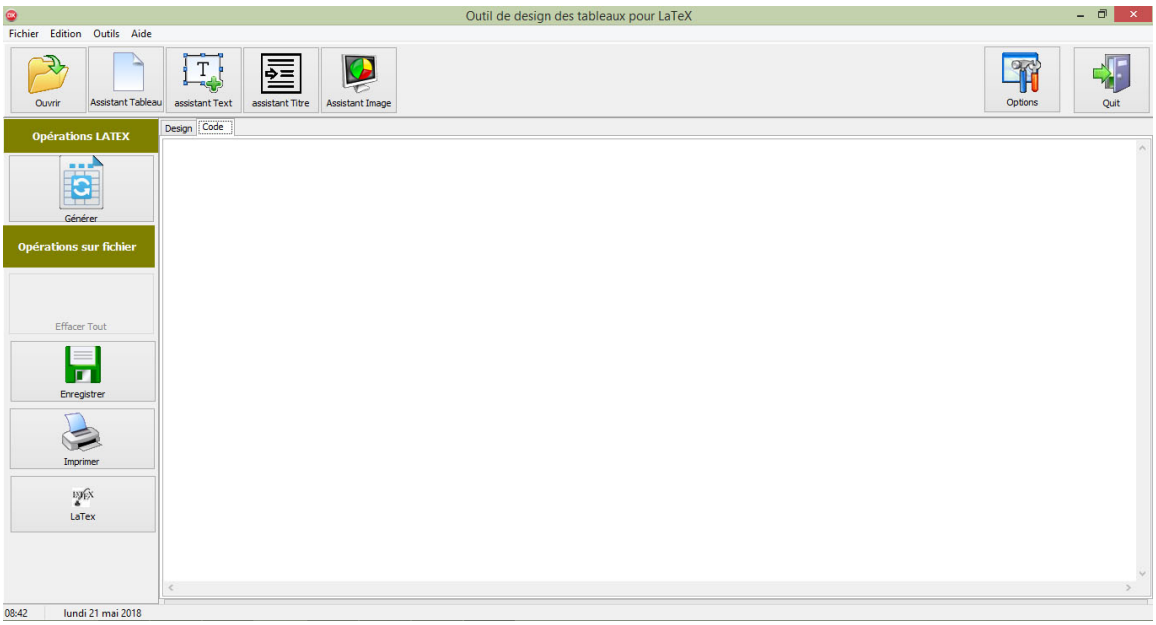

Figure 3.1: Interface principale de l'application

3.4.2 Fenêtre Assistant Tableau

Avec cette interface, l'utilisateur peut simplement créer un nouveau tableau en spécifiant le nombre de lignes et de colonnes, ajouter du texte à une cellule et exécuter toutes les autres options disponibles.

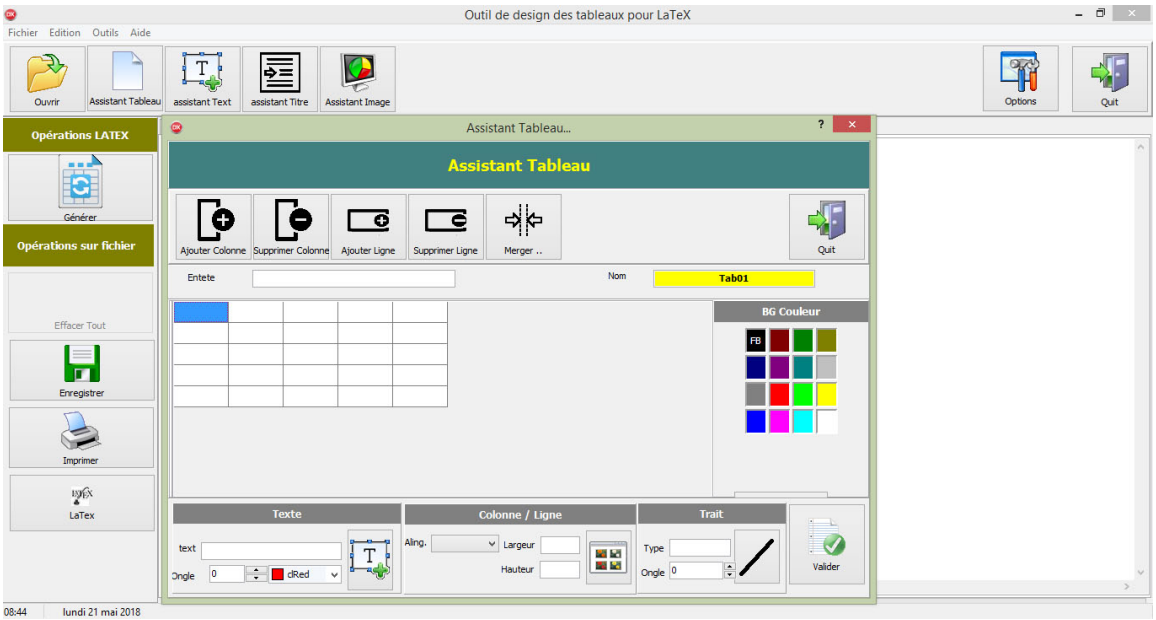

Figure 3.2: interface Assistante Tableau

3.4.3 Opérations sur un tableau

Pour tout changement dans la table d'ajouter ou de supprimer soit un ligne ou colonne, fusionnement ou fractionnement , un seul double clic sur le tableau qui mène l'utilisateur à la fenêtre d'assistant création de table actuelle, grâce à cette interface, il peut facilement modifier la table.

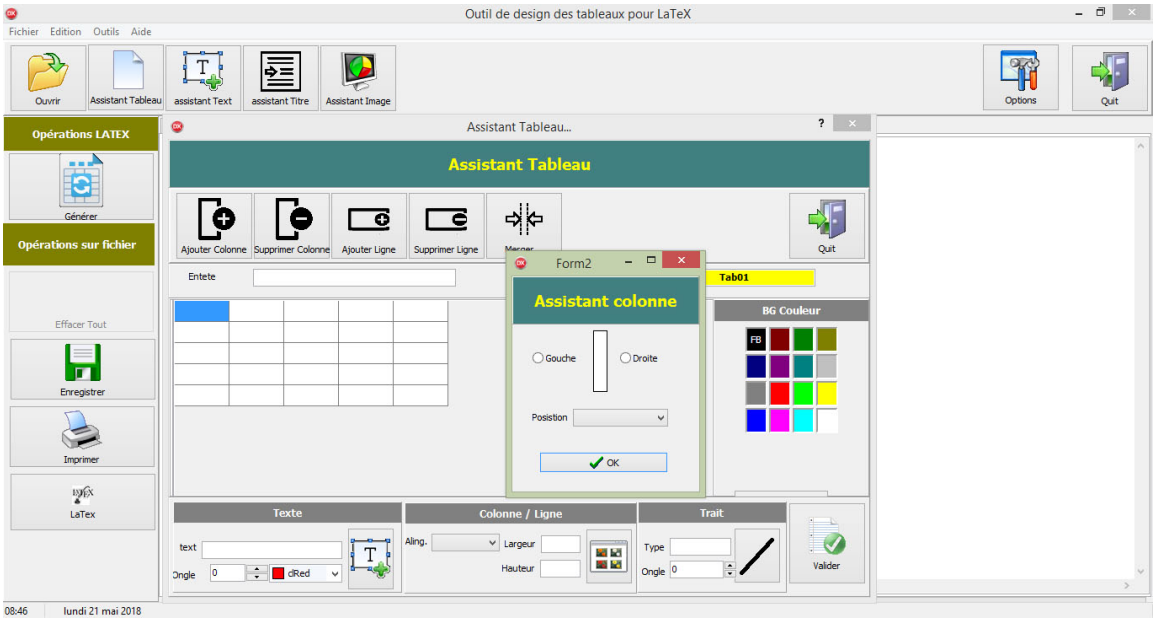

Figure 3.3: Interface Ajouter un colonne

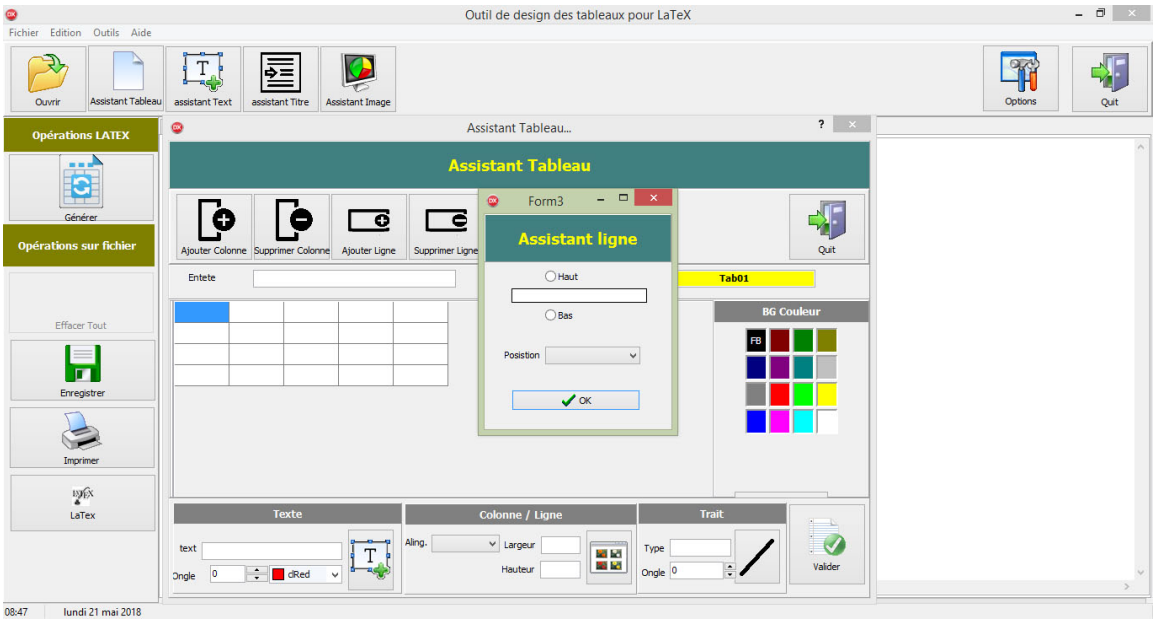

Figure 3.4: Interface Ajouter un ligne

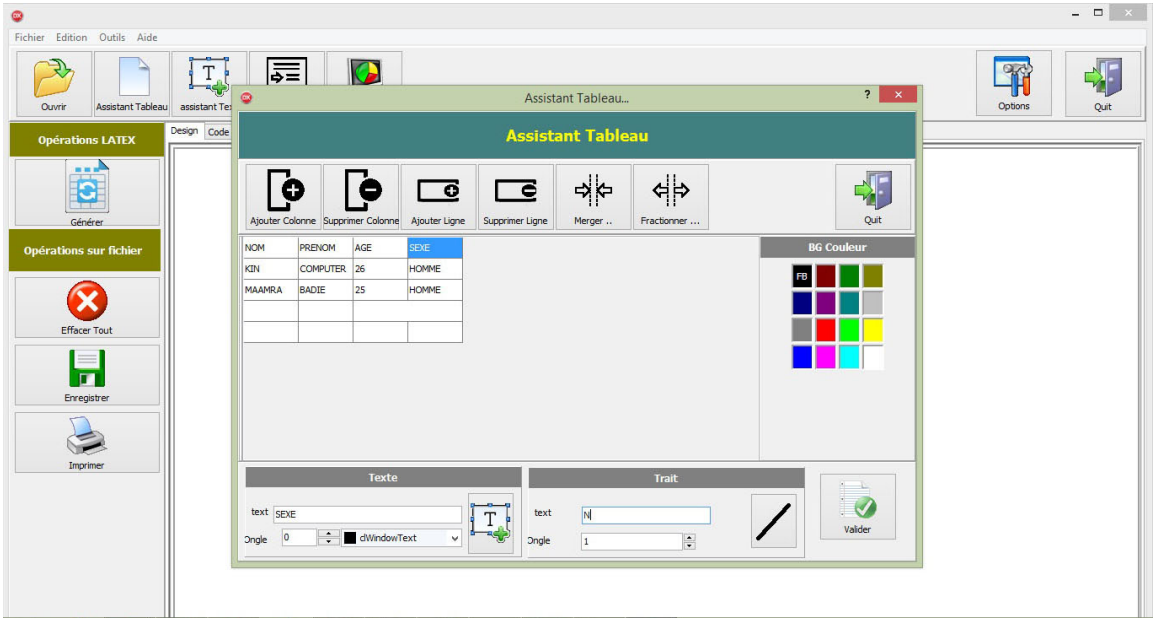

Figure 3.5: Interface Fusionnement cellule et Ajouter une Texte

3.4.4 Générer Le Code LaTeX

L'utilisateur Après avoir créé un tableau et l'avoir modifié selon son choix, peut le générer et obtenir leur code LaTeXdans un fichier externe de type .fk dans un répertoire voulu.puis il peut ouvrir ce fichier ultérieurement

L'utilisateur peut également voir le code résultant directement sur l'interface principale, sur la deuxième onglet de l'interface.il peut copier le code générer et l'utiliser en LaTeX.

En fin l'utilisateur peut générer directement leur document LaTeXcontient le tableau qui déjà éditer.

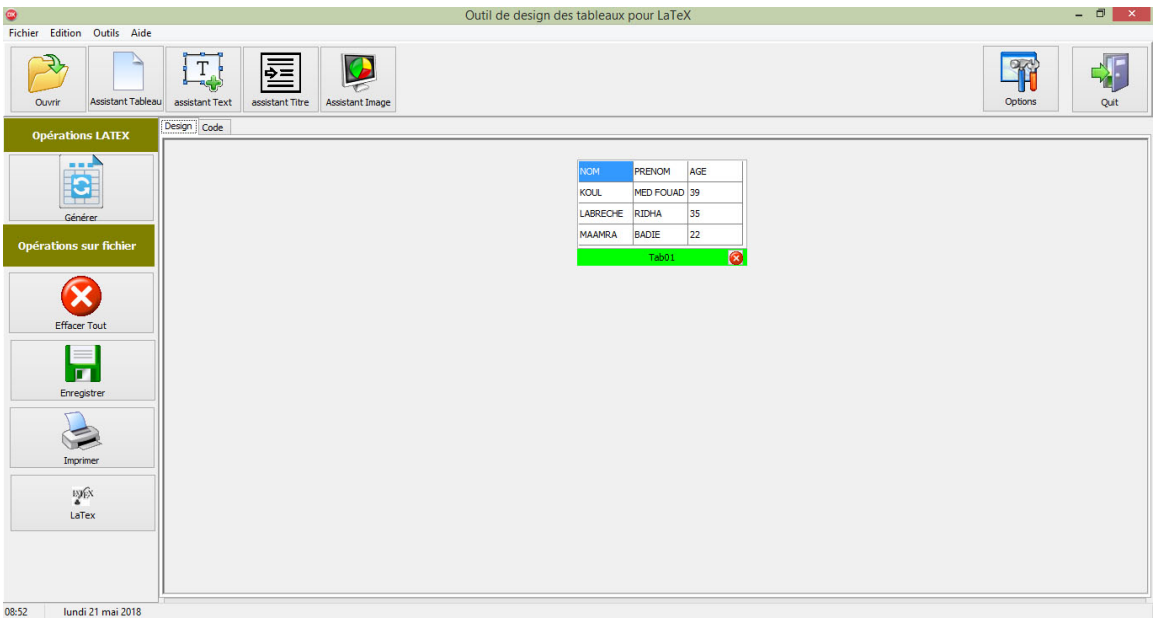

Figure 3.6: Interface ongle dessin de Tableau

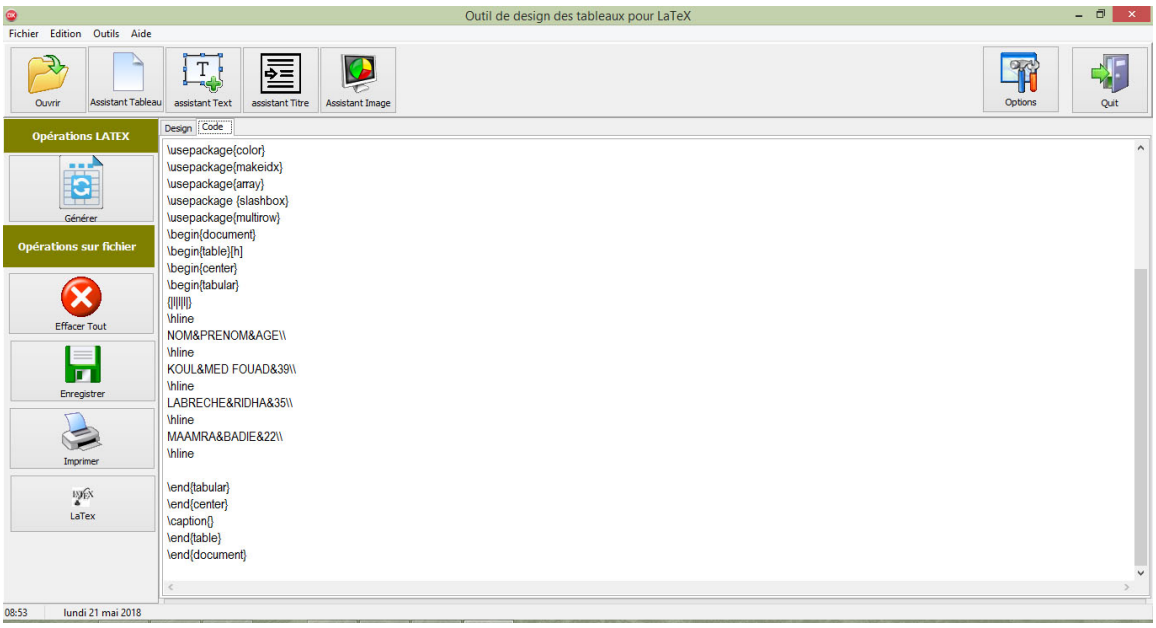

Figure 3.7: Interface ongle Génération Code LaTeX

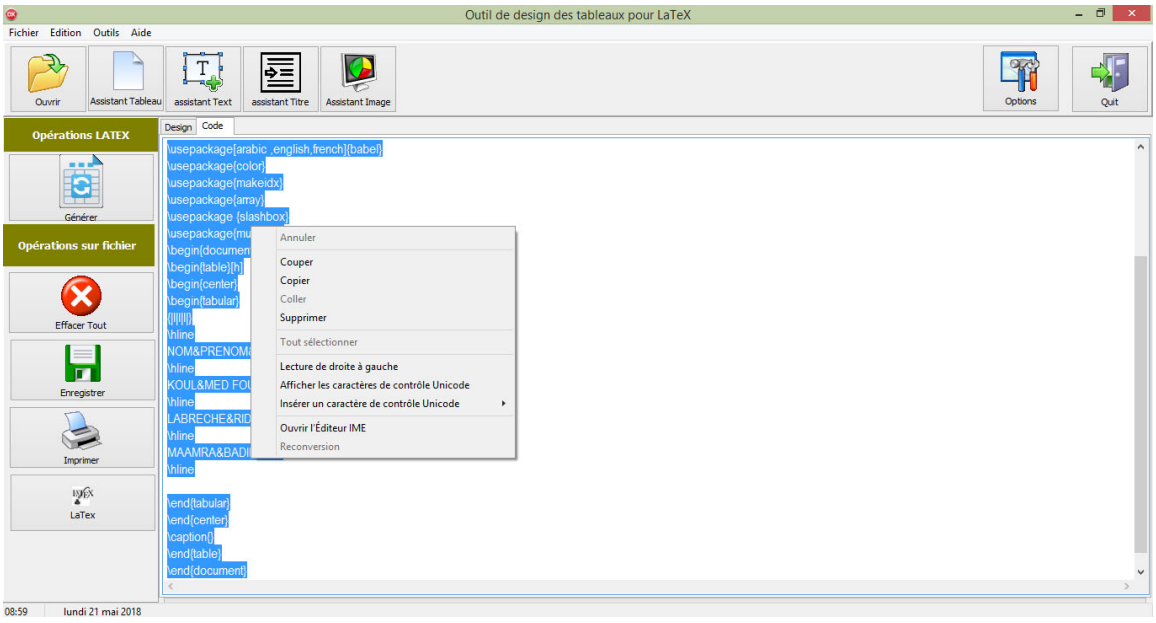

Figure 3.8: Interface copier un code LaTeX

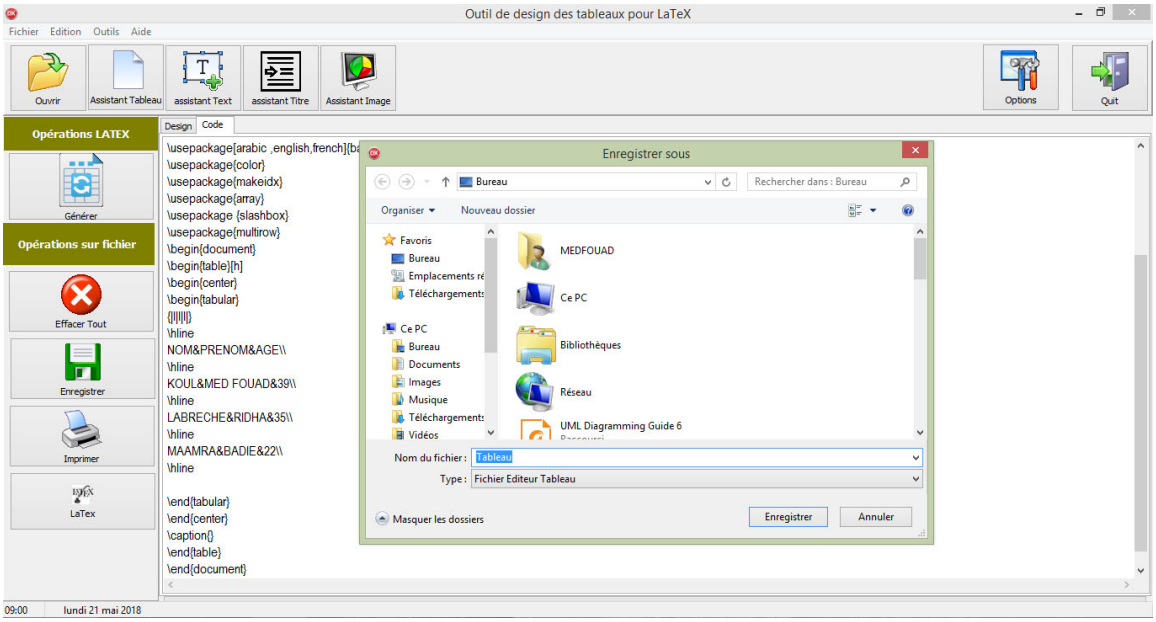

Figure 3.9: Interface d'enregistrement d'un code LaTeX

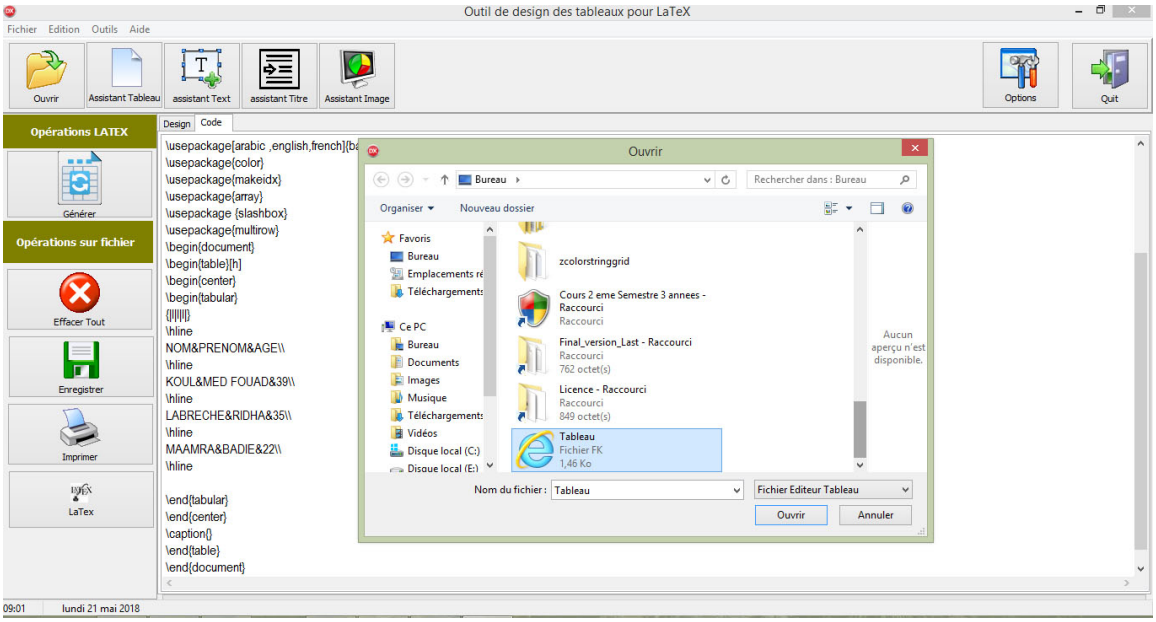

Figure 3.10: Interface Ouvrir Un fichier Tableau déjà enregistré

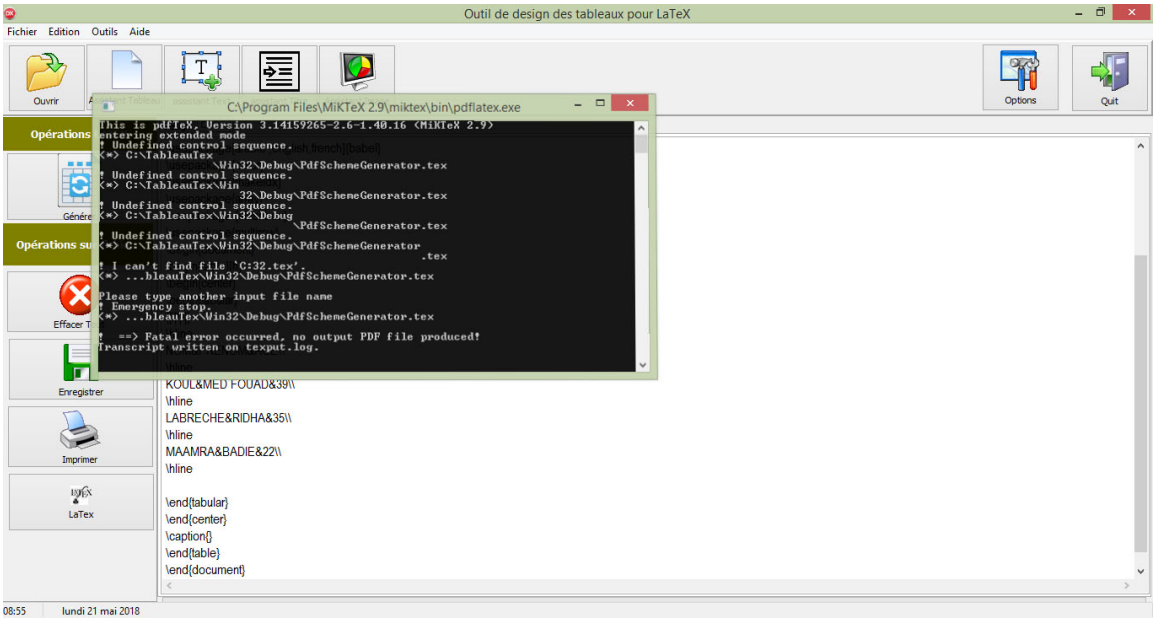

Figure 3.11: Génération Fichier Latex

Conclusion

Dans ce chapitre nous avons présenté comment nous avons développer l'application, ainsi que le langages utilisé et l'environnement de travail choisi pour atteindre ce résultat, puis nous avons présenté les codes et les interfaces principales et les menus de notre application.

Conclusion

Le travail que nous avons réalisé, dans ce rapport, consiste à développer une application, sous l'environnement Delphi, qui permet de générer le code LaTeX d'un tableau réalisé à l'aide d'une interface graphique. Cette application permet de créer des tableaux et de les modifier dans diverses opérations telles que la fusion des cellules, le fractionnement d'une cellule, l'ajout de colonnes et de lignes, l'édition d'un texte ainsi que d'autres opérations pour générer le code LaTeX à partir de l'interface graphique.

Ce travail nous a donné l'opportunité de savoir comment créer une interface entre deux applications informatiques (Delphi et Latex). Il nous a aussi ouvert une fenêtre pour prendre connaissance du logiciel de traitement de texte scientifique, tel que LaTeX. Ce dernier est un outil très utile pour les chercheurs surtout dans la rédaction de leurs travaux de recherche. Ainsi le but principal de ce mémoire a été réalisé permettra aux utilisateurs du Latex d'avoir un outil qui permet d'accélérer leur productivité et d'une utilisation plus ergonomique grâce à une interface graphique simplifiée. En conclusion, nous comptons que ce modeste travail sera un document de base pour les étudiants de la génération qui suit, afin qu'ils puissent approfondir ce thème pour traiter des objets LaTeX plus compliqués.

Bibliographie

[1] http://www.ordinateur.cc, "Logiciel de traitement de texte." http://www.ordinateur.cc/Logiciel/Logiciel-de-traitement-de-texte/180835.html, mar 2018. [Online; accessed 10 march 2018].

[2] A. Prikaznov, "Wysiwyg and wysiwym editors." https://dzone.com/articles/wysiwyg-and-wysiwym-editors, oct 2013. [Online; accessed 20 march 2018].

[3] A. LAZREK, "Présentation générale de latex," *Université Cadi Ayyad, Faculté des Sciences Département Informatique Marrakech - Moroc*, 2008.

[4] E. Guichard, "Latex, tout un programme...," *ENSSIB*, 2009.

[5] V. Lozano, *Tout ce que vous avez toujours voulu savoir sur LaTeX sans jamais oser le demander 1.0: ou comment utiliser LaTeX quand on n'y connait goutte*. Framabook, 2013.

[6] D. Bitouzé, "Installation de (la)tex," *Laboratoire de Mathématiques Pures et Appliquées Joseph Liouville et IUT Génie Thermique et Énergie de Dunkerque*, 2013.

[7] S. L. (Nikopol), "Les tableaux." https://latex.developpez.com/faq/?page=Les-tableaux. dernière mise à jour : 31 août 2017.

[8] C. B. Denis Conan, Chantal Taconet, "Introduction au langage de modélisation uml," *TELECOM SudPARIS*, oct 2015.

[9] P. Ciancarini, "Uml basic," *Aristotle, De Anima!*, p. 14.

[10] Yukikoi, "10.2 tokyo - release 2." http://docwiki.embarcadero.com, feb 2018. Dernière modification de cette page le 5 février 2018.