

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université des Sciences et de la Technologie d'Oran
Mohamed Boudiaf



Faculté d'Architecture et de Génie Civil
Département d'Architecture

Polycopié

Le dessin paramétrique appliqué à l'Architecture

Préparé et présenté par

Farid RAHAL

Année universitaire 2019 - 2020

Résumés

ملخص

الرسم الحدودي هو أساس المقاربة الحدودية للتصميم المعماري والتي تهدف إلى مساعدة المهندسين المعماريين في توليد أنواع معقدة من الأشكال والأحجام

بالإضافة إلى ذلك ، فإن الحاجة المتزايدة للتحكم في عدد كبير من المعلمات ومنحها المرونة و القدرة على التكيف في مجال الهندسة المعمارية تتطلب استخدام معايير ضخمة للرسومات التي سيتم إنشاؤها

يتيح التحالف بين قواعد البيانات وأدوات الرسم بمساعدة الكمبيوتر تحقيق هذا الهدف بفضل قدرة قواعد البيانات على إدارة كمية كبيرة من المعلومات وتحديثها بسرعة وبفضل تقنيات الرسم الحدودي. قادر على معالجة هذه البيانات وترجمتها إلى رسم وكائن مشروع

يتيح استخدام العشوائية في مجموعة قيم المعلمات المختلفة استكشاف حلول غير متوقعة. ومع ذلك ، فإن الصدفة مقيدة بالشروط التي يفرضها المهندس المعماري حتى يتمكن من توجيه المشروع في الاتجاه الذي يرغب فيه وبالتالي ، يمكن أن توفر تقنيات الرسم الحدودي حلولاً مبتكرة لمشاكل التصميم المعماري.

الكلمات المفتاحية: رسم حدودي ، قاعدة بيانات ، رسم بمساعدة الحاسوب ، تصميم معماري

Résumé.

Le dessin paramétrique est la base de l'approche paramétrique de la conception architecturale qui est destinée à aider les architectes dans la génération de types complexes de formes et de volumes.

En outre, le besoin croissant de maîtriser un grand nombre de paramètres et de leur donner souplesse et adaptabilité dans le domaine de l'architecture nécessite l'utilisation d'un paramétrage massif des dessins à générer.

L'alliance entre les bases de données et les outils du dessin assisté par ordinateur permet d'atteindre cet objectif grâce à la capacité des bases de données à gérer et mettre à jour rapidement une grande quantité d'informations et grâce aux techniques du dessin paramétrique pouvant manipuler ces données et les traduire en dessin et en objet du projet.

L'utilisation de l'aléatoire dans la combinaison des valeurs des différents paramètres permet d'explorer des solutions insoupçonnées. Cependant, le hasard est contraint par les conditions imposées par l'architecte pour qu'il puisse diriger le projet dans la direction qu'il souhaite.

Ainsi, les techniques du dessin paramétrique peuvent apporter des solutions novatrices aux problèmes de conception architecturale.

Mots clés : Dessin paramétrique, Base de données, Dessin assisté par ordinateur, Conception architecturale.

Abstract

Parametric drawing is the basis of the parametric approach to architectural design which is intended to assist architects in the generation of complex types of shapes and volumes.

In addition, the growing need to control a large number of parameters and to give them flexibility and adaptability in the field of architecture requires the use of massive parameterization of the drawings to be generated.

The alliance between databases and computer-aided drawing tools makes it possible to achieve this objective thanks to the ability of databases to manage and quickly update a large amount of information and thanks to parametric drawing techniques. able to manipulate this data and translate it into a drawing and a project object.

The use of randomness in the combination of the values of the various parameters makes it possible to explore unsuspected solutions. However, chance is constrained by the conditions imposed by the architect so that he can direct the project in the direction he wishes.

Thus, parametric drawing techniques can provide innovative solutions to architectural design problems.

Keywords: Parametric drawing, Database, Computer assisted drawing, Architectural design.

Tables des matières

Tables des matières

Introduction générale	14
Chapitre I : Introduction au dessin paramétrique.....	18
I.1. Introduction.....	19
I.2. Approche historique.....	19
I.3. Tracés et procédés anciens.....	21
I.4. Evolution de l'approche paramétrique en Architecture.....	22
I.5. L'avènement du numérique.....	22
I.6. Conclusion	23
Chapitre II : Le dessin technique.....	25
II.1. Introduction.. ..	26
II.2. Les bases du dessin technique.....	26
II.2.1. Echelles.....	26
II.2.2. Les formats de représentation du dessin technique.....	28
II.2.3. Les projections.....	29
II.3. Les coupes.....	31
II.4. Conclusion.....	32
Chapitre III : Le dessin assisté par ordinateur.....	33
III.1. Introduction.....	34
III.2. Définitions.....	34
III.3. Les objectifs du dessin assisté par ordinateur.....	34
III.4. Les principes de base du dessin assisté par ordinateur.....	36
III.5. Les avantages du dessin assisté par ordinateur.....	36
III.6. Les logiciels du dessin assisté par ordinateur.....	37
III.5. Conclusion.....	38
Chapitre IV : Les principes de base du dessin paramétrique.....	39
IV.1. Introduction	40
IV.2. Du dessin assisté par ordinateur au dessin paramétrique.....	40
IV.3. Le rôle de la programmation dans le dessin paramétrique.....	42
IV.4. Le dessin aléatoire.....	43
IV.5. Conclusion.....	46
Chapitre V : Notions de programmation.....	47
V.1. Introduction.....	48

V.2.Le développement d'un programme.....	49
V.3. Les langages de programmation.....	49
V.4. La notion d'algorithme.....	50
V.4.1. Historique.	50
V.4.2. Définition	50
V.4.3. Liste d'instructions.....	51
V.4.4. Les caractéristiques d'un algorithme.....	51
V.5.Comment programmer ?.....	53
V.6. Diffusion des logiciels.....	57
V.7. Conclusion.....	57
Chapitre VI : Les bases de données.....	59
VI.1. Introduction.....	60
VI.2. Définition d'une base de données	60
VI.3. Utilité d'une base de données.....	61
VI.4. La gestion des bases de données.....	62
VI.5. Les principaux SGBD.....	64
VI.6. Les caractéristiques d'un SGBD.....	64
VI.7. Les modèles de base de données.....	65
VI.8. Exemple de base de données.....	65
VI.9. Applications.....	70
VI.10. Conclusion.....	75
Chapitre VII : Les bases de données et le dessin paramétrique.....	77
VII.1. Introduction.....	78
VII.2. Liaison de la base de données et de l'outil de DAO.....	78
VII.3. Partage des données.....	80
VII.4. Application.....	81
VII.5. Conclusion.....	84
Chapitre VIII : La démarche BIM.....	86
VIII.1. Introduction	87
VIII.2. Les objectifs du BIM.....	88
VIII.3. Les avantages du BIM.....	89
VIII.4. Les outils propriétaires du BIM.. ..	90

VIII.5. Le format IFC.....	92
VIII.6. Les outils libres du BIM.....	92
VIII.7. Une autre alternative aux outils BIM propriétaires.....	93
VIII.8. Conclusion.....	95
Conclusion générale.....	97
Références bibliographiques.....	100

Liste des figures

Figure I.1. Underweysung der Messung, Instructions sur la mesure par Dürer en 1525.....	20
Figure I.2. Instrument intégrant les variations nécessaires au calcul du rapport de proportion	21
Figure I.3. Centre Heydar Aliyev à Bakou, en Azerbaïdjan, par Zaha Hadid.....	23
Figures II.1. Les formats de représentation du dessin technique.....	28
Figures II.2. Cube de projection.....	29
Figures II.3. Façade principale.....	30
Figures II.4. Coupe AA.....	31
Figure III.1. Domaines d'applications du dessin assisté par ordinateur.....	35
Figure III.2. Exemples de commandes sous Autodesk-Autocad.....	38
Figure IV.1. Exemple d'un dessin en 2 dimensions.....	40
Figure IV.2. Un dessin en 3 dimensions extrudé à partir d'un autre dessin en 2 dimensions.....	41
Figure IV.3. Variante en 2 dimensions de la figure IV.1.....	41
Figure IV.4. Exemple d'un algorithme de dessin et le résultat obtenu.....	42
Figure IV.5. Formulaire permettant la saisie des valeurs des paramètres et l'exécution.....	43
Figure IV.5. Formulaire et résultat d'un dessin aléatoire.....	44
Figure IV.5. 2000 segments dessinés aléatoirement mais contraints par des limites.....	45
Figure IV.7. Segments et hachures générées aléatoirement dans des limites prédéfinies.....	45
Figure VI.1 : Architecture clients-serveur et architecture trois tiers des bases de données.....	61
Figure VI.2. Les composants d'un système de gestion des bases de données.....	63
Figure VI.3. Structure de la table commune.....	70
Figure VI.4. Contenu de la table commune.....	70
Figure VI.5. Formulaire pour la gestion des données de la table commune.....	71
Figure VI.6. Requête sur la table commune.....	71
Figure VI.8. Formulaire de la base de données pour la gestion des grilles.....	73
Figure VI.9. Catégorisation des espaces supportés par des grilles régulatrices	74
Figure VI.10. Structure et liaisons des tables de la base de données.....	75
Figure VII.1. Liaison des bibliothèques d'Autodesk-Autocad sous MS-Access.....	79
Figure VII.2. Structure de table, formulaire dédié et résultat sous forme de dessin.....	80
Figure VII.3. Objets réalisés par plusieurs intervenants à travers un réseau local.....	81
Figure VII.4. Tables de la base de données dédiée à la gestion des espaces.....	82
Figure VII.5. Formulaire pour la gestion des données et des dessins liés aux espaces conçus.....	83
Figure VII.6. Le résultat de la simulation sous forme de dessin, de plan et de volume.....	84
Figure VIII.1. Exemple d'un projet réalisé avec un outil BIM propriétaire.....	87
Figure VIII.2. Le cycle de vie de la démarche BIM.....	88
Figure VIII.3. Les avantages du BIM.....	89
Figure VIII.4. Quelques outils BIM propriétaires.....	91
Figure VIII.5.. Le système d'abonnement d'un outil BIM propriétaire.....	91
Figure VIII.6. Quelques outils BIM libres.....	93
Figure VIII.7. L'architecture du système d'information d'un projet lié à un outil de CAO/DAO.....	94
Figure VIII.8. Exemple d'un projet conçu à l'aide d'une base de données lié à un outil de CAO/DAO.....	95

Liste des tableaux

Tableau II.1. Exemples d'échelles de représentation du dessin technique.....	27
Tableau II.2. Les utilisations de chaque type d'échelle.....	27
Tableau II.2. Les formats de représentation du dessin technique.....	28
Tableau VI.1. Les types de données dans une base de données.....	67

Introduction générale

La conception architecturale voit le recours de plus en plus aux formes libres ou non standard. Ce phénomène ne peut exister que sous l'influence des technologies numériques, et plus particulièrement des outils paramétriques d'aide à la conception et au dessin.

Un dessin classique, qui peut être un plan d'architecture, est un ensemble d'entités simples sous forme de segments et d'arcs interdépendants dans leurs positions relatives et leurs conditions géométriques. Cette interdépendance peut être traduite par des équations mathématiques qui sont transparentes à l'utilisateur dans les outils conventionnels.

Dans ce type d'équation, le résultat dépend des valeurs affectées aux variables. Ainsi, dans un outil de dessin paramétrique en deux dimensions, lorsque l'on change la dimension d'un élément existant ou sa position angulaire, ou n'importe quel autre paramètre le définissant, cet élément est immédiatement recalculé par le logiciel qui le redessine alors en fonction des changements, tout en modifiant les autres éléments qui lui sont reliés, car ceux-ci sont contraints de respecter les conditions qui ont présidé à leur mise en place initiale.

Ces intéressantes caractéristiques du dessin paramétrique ont fait qu'il soit le cœur de l'architecture paramétrique permettant de concevoir, de modéliser ou de dessiner de manière, certes plus complexe que les approches classiques mais cette méthode donne au projet architectural flexibilité et adaptabilité vis-à-vis de différents types de contraintes liées à l'environnement, au fonctionnement ou aux risques.

Ce type d'architecture n'aurait pas pu voir le jour sans l'avènement de la révolution numérique qui a profondément bouleversé notre société, notre mode de vie et notre façon de travailler.

En effet, l'essor des techniques numériques s'est traduit par une capacité de calcul et de stockage des données du projet plus accrue, mais aussi par une potentielle mise en réseau planétaire des acteurs de cette récente approche de l'Architecture. Ainsi, la circulation des idées s'est décentralisée, d'où un travail collaboratif plus efficace.

Ainsi, le projet conçu à partir des techniques de l'architecture paramétrique pourra murir dans les ordinateurs des collaborateurs à partir de paramètres liés à la structure, à l'environnement, aux matériaux, ou aux systèmes de fabrication.

Les outils de dessin et de modélisation paramétriques offrent un degré de flexibilité qui n'est pas possible avec les outils conventionnels. L'approche paramétrique permet de modifier facilement les éléments complexes sans avoir à les reconstruire.

De cette manière, les paramètres utilisés pour créer un objet peuvent être modifiés par la suite pour obtenir un résultat différent sans reconstruire manuellement l'élément. Des modifications peuvent être apportées aux éléments qui ne sont jamais figés sauf à être totalement contraint, les autres éléments de l'ensemble sont également modifiés en conséquence.

L'utilisation des bases de données pour le recueil des paramètres du dessin permet l'exploitation de la pleine puissance de cet outil. En effet, le paramétrage massif d'un projet ne peut être géré correctement qu'avec une base de données.

Si celle-ci est en partage, elle permettra le travail collaboratif entre les différents intervenants du projet pour sa réalisation. Ce qui représente les grands principes de la démarche BIM (Building information modelling).

Il est ainsi possible de faire du BIM sans avoir recours aux outils propriétaires dont l'acquisition pour un travail professionnel rencontre bien souvent une barrière économique dans les pays en voie de développement.

Les bases de données partagées et liées aux techniques du dessin paramétriques, offrent une alternative intéressante aux outils BIM classiques.

Ce polycopié, adressé aux étudiants de Licence 3 en Architecture, présente l'ensemble des notions nécessaires à la bonne compréhension du dessin paramétrique et de son utilisation efficace dans le domaine de l'Architecture.

Le polycopié est divisé en huit chapitres. Le premier chapitre est une introduction au dessin paramétrique qui est une évolution marquante du dessin technique dont le deuxième chapitre y est dédié.

Le troisième chapitre apporte un éclairage sur le dessin assisté par ordinateur et ses différentes notions. Le quatrième chapitre donne les principes de base du dessin paramétrique dont la maîtrise ne peut se faire sans un savoir-faire avéré en programmation que le chapitre cinq en donne les notions qui permettent de manipuler les données mais aussi les objets du dessin.

Ces données devront être stockées, traitées et analysées. Ce qui peut être réalisé par le biais des bases de données auxquelles le chapitre six est dédié.

Le chapitre sept s'intéresse à la combinaison des bases de données et des techniques du dessin paramétrique. Ce qui correspond aux impératifs de la démarche BIM à laquelle le huitième et dernier chapitre est consacré.

Une conclusion générale clôt ce polycopié en présentant les points forts du dessin paramétriques et de la méthodologie présentée et posant les jalons d'une évolution future qui pourrait se refléter dans les travaux des étudiants qui seront les architectes de demain.

Chapitre I

Introduction au dessin paramétrique

I.1. Introduction.

Le dessin paramétrique est une fonctionnalité attrayante par rapport au dessin classique. Il est ainsi possible de définir des relations persistantes entre les objets d'un projet. Une fois ces relations établies, elles seront appliquées tout au long de l'évolution du dessin.

Cette fonctionnalité peut avoir de multiples applications mais également paraître intimidante ou compliquée au premier abord. Il est alors essentiel d'en comprendre les fondements et ses évolutions historiques.

I.2. Approche historique.

L'approche paramétrique est en réalité fondamentalement présente au coeur de l'histoire de l'architecture (Delvaux et al., 2016). L'architecte Bernard Cache dans son exposé «Dürer - Vitruvius - Plato. Instruments of Thought » (Cache, 2012), celui-ci propose une lecture paramétrique de l'Underweysung der Messung de Albrecht Dürer, pour mettre en évidence plusieurs aspects paramétriques intrinsèques à la nature de certains édifices dès l'Antiquité. Cet ouvrage de Dürer, écrit en 1525, s'attache à décrire pour un usage opérationnel des procédés et leur instrumentation. Ses fins consistent à la conception d'objets ou d'éléments architecturaux dont la variation est une caractéristique intrinsèque comme par exemple les colonnes torsées, l'inclinaison ou le fruit des murs de fortification.

L'intérêt majeur de cette approche réside dans deux de ses aspects:

- faire apparaître la présence historique d'une tradition de la variation et de la mécanique dans la conception d'objets architecturaux
- révéler les structures logiques qui préfigurent celles que nous utilisons aujourd'hui avec les logiciels de conception paramétrique (Cache, 2013).

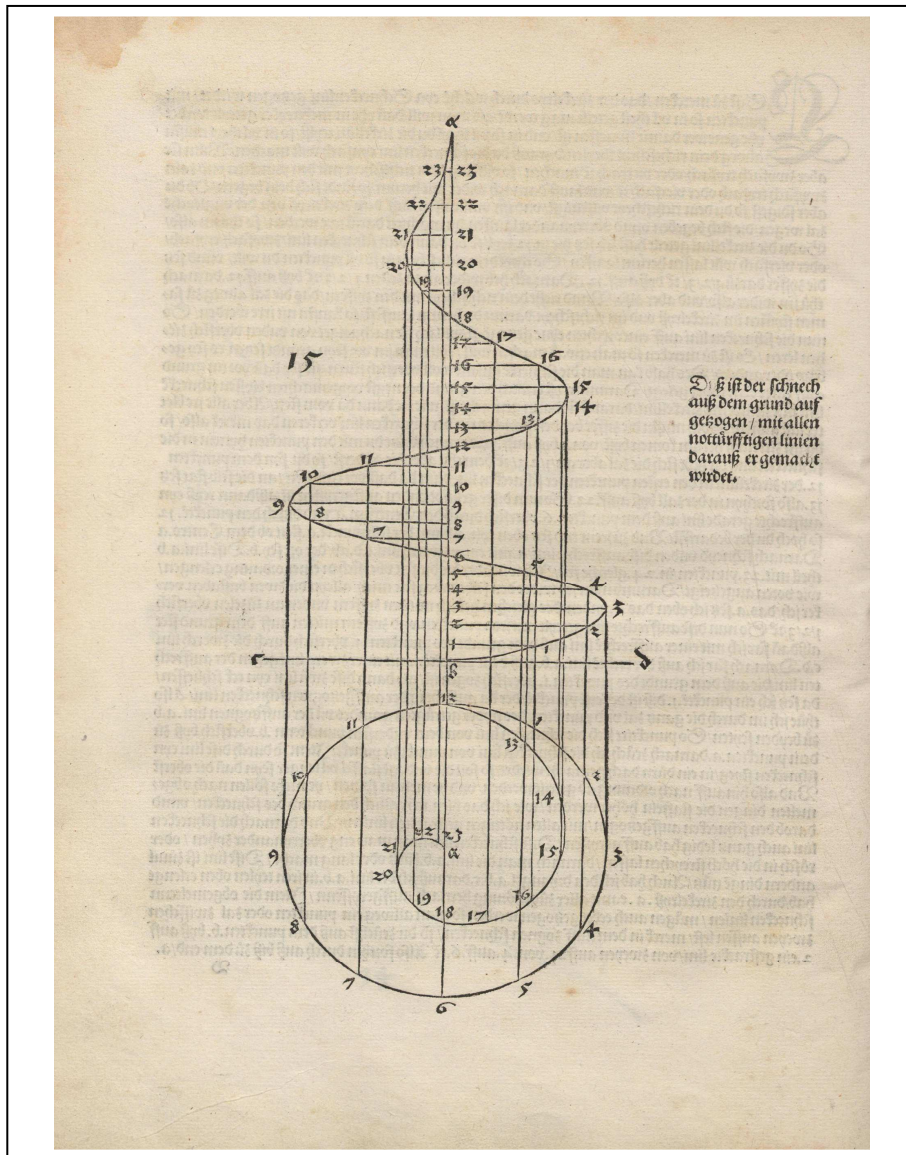


Figure I.1. Underweysung der Messung, Instructions sur la mesure par Dürer en 1525.

En effet, Dürer développe en quatre livres les principales constructions géométriques comme la spirale d'Archimède, la spirale logarithmique, la conchoïde, l'épicycloïde, le limaçon de Pascal, des constructions approchées des polygones réguliers à 5, 7, 9, 11 ou 13 côtés et de la trisection de l'angle et de la quadrature du cercle, des constructions de solides géométriques (cylindre, solides de Platon), une théorie de l'ombre et de la perspective.

La géométrie descriptive qui est à l'origine de la morphométrie et qui est nécessaire à la représentation des corps dans l'espace, initiée par Dürer sera reprise, deux siècles plus tard, par Gaspard Monge qui en fera un développement complet et artistique.

I.3. Tracés et procédés anciens.

La lecture paramétrique des travaux de Dürer met en lumière les procédés qu'il a utilisé afin d'arriver à paramétrer ses ouvrages.

En effet, il a substitué un instrument numérique à un instrument mécanique. Il a mis en place un procédé qui consiste en un couplage de valeurs numériques sous forme d'angles et de longueurs qui sont variables. Ainsi, il a pu organiser la variation numérique des paramètres de son ouvrage (Delvaux et al., 2016).

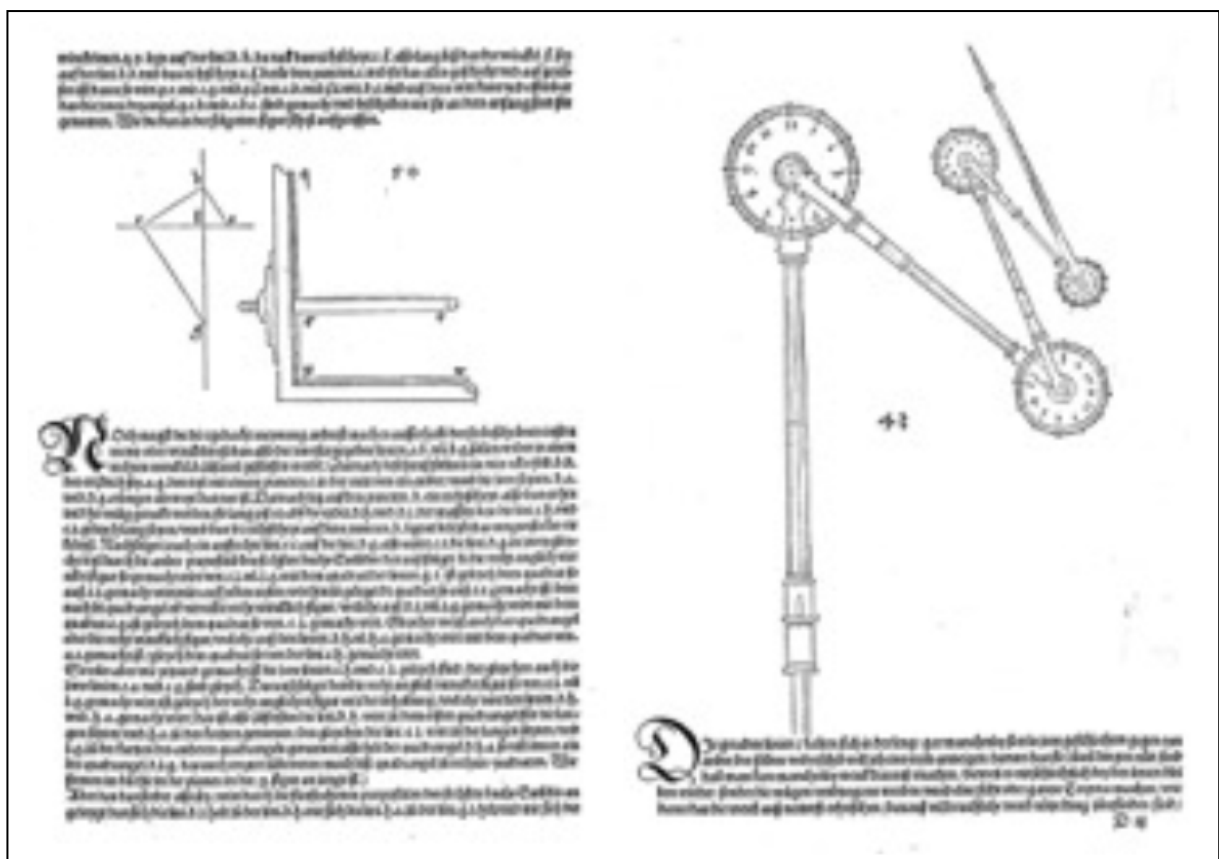


Figure I.2. A gauche: instrument intégrant les variations nécessaires au calcul du rapport de proportion continue. A droite: mesures pour le développement d'une ligne serpentine régulière. Il s'agit d'une variation combinée de mouvements angulaires et de distances.

Source : Delvaux et al., 2016

La conception et le dessin paramétrique sont bien ancrés dans l'histoire de l'architecture. Les œuvres d'architectes et d'ingénieurs comme les prémisses de Shukov ou Gaudi, les développements de Candela, de Dieste et les logiques fondatrices des projets de Luigi Moretti ou S. Musmeci, en sont le reflet.

I.4. Evolution de l'approche paramétrique en Architecture.

L'Architecte italien Luigi Moretti qui a à son actif le fameux quartier du Watergate à Washington et surtout l'Hôtel Aurassi à Alger, avait fondé en 1957 un groupe de recherche, le « Institute for Operations Research and Applied Mathematics Urbanism » dont la production sera exposée en 1960 à Milan lors de l'Exposition Triennale, sous l'intitulé « Exhibition of Parametric Architecture and of Mathematical and Operational Research in Town Planning ».

La conception et le dessin paramétrique en architecture selon Luigi Moretti est identifié à une pratique de la discipline qui tend à introduire les phénomènes de l'architecture et de l'urbanisme dans les structures de pensée contemporaines, et en particulier scientifiques (Bravo, 2014). Malgré l'aspect novateur et interdisciplinaire de la démarche, elle ne semble toutefois pas avoir laissé un héritage marquant dans la connaissance de l'architecture (Delvaux et al., 2016).

I.5. L'avènement du numérique.

Le blocage qu'a connu la démarche de Luigi Moretti n'a pu être levé qu'avec l'avènement des technologies numériques basées sur des ordinateurs de plus en plus puissants permettant de manipuler des données massives avec comme conséquence l'apparition de géométries innovantes telle qu'elles apparaissent dans les œuvres de Zaha Hadid.



Figure I.3. Centre Heydar Aliyev à Bakou, en Azerbaïdjan, par Zaha Hadid.

I.6. Conclusion.

Les fondements de la modélisation paramétrique intrinsèque à des modes de conception ancrés dans une tradition méthodologique remontent à l'Antiquité. (Delvaux et al., 2016). L'approche paramétrique en Architecture consiste à choisir un certain nombre de paramètres indépendants les uns des autres et à les soumettre à des variations systématiques. Ces variations sont opérées en fonction de critères définis afin d'obtenir non pas un objet, mais une série d'objets (Jormakka, 2008).

Ce principe a été largement appliqué dans la conception assistée par ordinateur, comme un ensemble de variables organisées en système, qui nourrissent des équations algorithmiques et lié à une géométrie résultant d'un processus.

L'approche paramétrique est aujourd'hui participative, collaborative, interactive, et collective en esprit (Bressani et al., 2019).

Avec l'essor du Big Data et des applications de l'apprentissage machine ainsi que de l'intelligence artificielle dans le domaine du dessin et de la conception architecturale, le processus paramétrique est devenu incontournable dans les projets où la complexité est avérée.

Malgré cette apparente complexité, le dessin paramétrique est basé essentiellement sur les fondements du dessin technique dont les principes de base sont présentés dans le chapitre suivant.

Chapitre II

Le dessin technique

II.1. Introduction.

L'apprentissage du dessin à main levée exerçait l'œil, du dessinateur qui faisait plus confiance à son regard affûté pour jauger d'une proportion ou d'un parallélisme. Le bras, la main et le crayon étaient le prolongement du cerveau. Le dessin était une pensée, une intelligence. (Charreb-Yssad, 2016). Le dessin technique est venu normaliser ce savoir-faire en y apportant la rigueur nécessaire pour y éliminer toute sorte d'ambiguïté et le faire adopter par la majeure partie des corps de métier.

Le dessin technique est un ensemble de conventions standardisées pour représenter des objets et des constructions sous forme de structures ou de bâtiments.

Ensemble, ces conventions constituent un langage visuel et permettent de s'assurer que le dessin est sans ambiguïté et relativement facile à comprendre. De nombreux symboles et principes du dessin technique sont codifiés dans une norme internationale appelée ISO 128.

Ces accords garantissent que le produit ou la construction montrés est tel qu'imaginé par le concepteur.

II.2. Les bases du dessin technique.

Le dessin technique est un langage qui permet d'optimiser la communication. Dans le domaine technique, le langage parlé n'est pas assez universel ni vraiment suffisant pour communiquer des idées et des modalités d'exécution. Le dessin technique est un langage figuratif pour la représentation, la communication technique, la conception et l'analyse systémique. Il s'agit d'un ensemble de conventions pour représenter des objets ; ces conventions assurent que l'objet produit est tel qu'il est imaginé par le dessin par son concepteur. Les différents types de dessins techniques sont :

- le croquis, généralement à main levée
- l'esquisse ou ébauche qui représente l'avant-projet
- l'épure : étude technique
- le schéma pour présenter le fonctionnement
- les dessins d'ensemble, dessin de définition,...

II.2.1. Echelles.

Lorsque les systèmes sont grands tels que les immeubles, les bateaux ou les automobiles ou bien petits comme les montres ou les circuits électroniques, il est nécessaire de faire des réductions ou des agrandissements pour les représenter.

Ci-dessous, quelques exemples d'échelles.

échelle	Sur le dessin...
1 : 1	1 cm correspond à 1 cm en réalité
1 : 2	1 cm correspond à 2 cm en réalité
2 : 1	2 cm correspondent à 1 cm en réalité

Tableau II.1. Exemples d'échelles de représentation du dessin technique.

Le type d'échelle est intimement lié à son utilisation. Le tableau II.2, donne une idée sur cet aspect.

<i>Echelle</i>	<i>Où l'utiliser ?</i>
<i>1 : 1000</i>	<i>Plans de situation et d'implantation</i>
<i>1 : 500</i>	
<i>1 : 200</i>	
<i>1 : 100</i>	<i>Dessins d'avant-projet</i>
<i>1 : 50</i>	<i>Dessins de projet</i>
<i>1 : 20</i>	<i>Dessins de détail d'un bâtiment</i>
<i>1 : 10</i>	
<i>1 : 5</i>	
<i>1 : 2</i>	

Tableau II.2. Les utilisations de chaque type d'échelle.

II.2.2. Les formats de représentation du dessin technique.

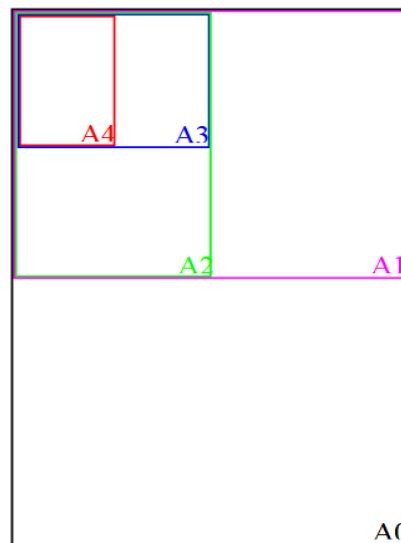
Il s'agit des formats normalisés issus du format de base dont la surface est :

$$1\text{m}^2 = 840 \text{ mm} \times 1.188 \text{ mm}.$$

Format	Dimensions
A 4	210x297 (mm)
A 3	420x297 (mm)
A 2	420x594 (mm) = 0,25 m ²
A 1	840x594 (mm) = 0,5 m ²
A 0	840x1188 (mm) = 1 m ²

Tableau II.2. Les formats de représentation du dessin technique.

Les autres formats sont obtenus par pliage.



Figures II.1. Les formats de représentation du dessin technique.

II.2.3. Les projections.

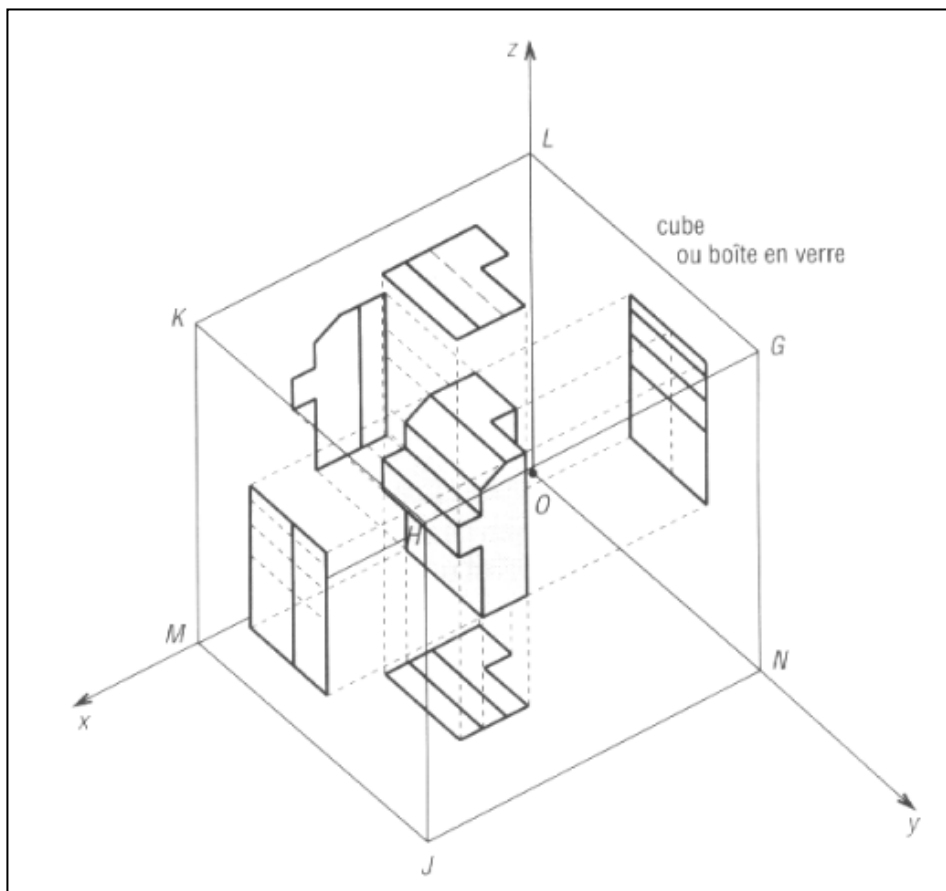
Une seule projection ne suffit pas pour représenter un objet, on utilise en dessin technique un ensemble de projections qui permettent une représentation non ambiguë.

Les représentations normalisées en dessin technique sont des projections orthogonales.

Les dessins à vues multiples, basés sur les méthodes de projections orthogonales, sont le mode de communication le plus usuel du monde industriel et technologique. Ces dessins servent à définir les formes, les contours, les dimensions et les bases de réalisation des pièces ou objets.

Les positions relatives des projections et leurs directions sont généralement normalisées dans un cube de projection.

L'observateur se place perpendiculairement à l'une des faces du système à définir. La face observée est ensuite projetée et dessinée dans un plan de projection parallèle à cette face, situé en arrière du système.

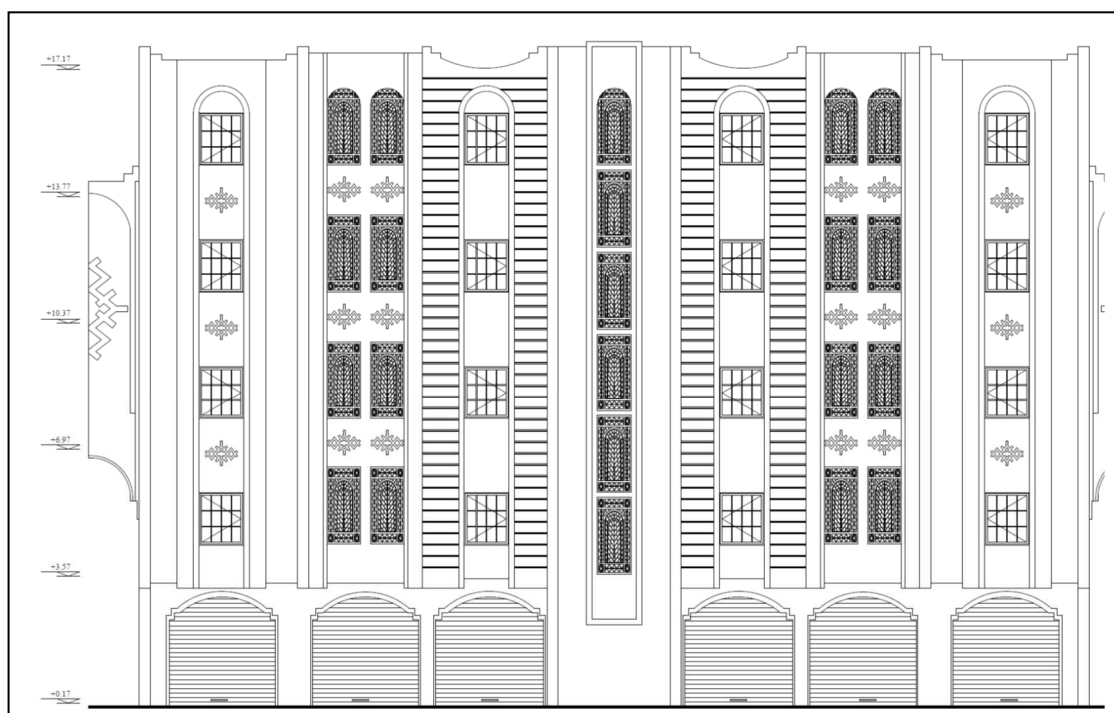


Figures II.2. Cube de projection.

Lorsque l'on déplie le cube, on identifie les vues et on remarque que selon la norme européenne :

- la vue de droite est placée à gauche de la vue de face,
- la vue de gauche est placée à droite de la vue de face,
- la vue de dessus est placée au-dessous de la vue de face,
- la vue de dessous est placée au-dessus de la vue de face
- la vue d'arrière est placée indifféremment à droite ou à gauche des vues.

Les vues retenues doivent toutes être correctement alignées et avoir une correspondance entre elles. La vue la plus caractéristique de l'objet à représenter est normalement choisie comme vue principale ou comme vue de face. En pratique, les six vues sont rarement utilisées et le nombre des vues retenues (vues normales, coupes ou sections) doit être limité à ce qui est nécessaire mais suffisant pour définir l'objet dans l'espace, sans ambiguïté, et éviter les répétitions inutiles de détails. La vue arrière est rarement utilisée. En règle générale, il faudrait choisir en priorité les vues donnant le maximum de clarté et présentant le moins de traits interrompus courts ou pointillés.



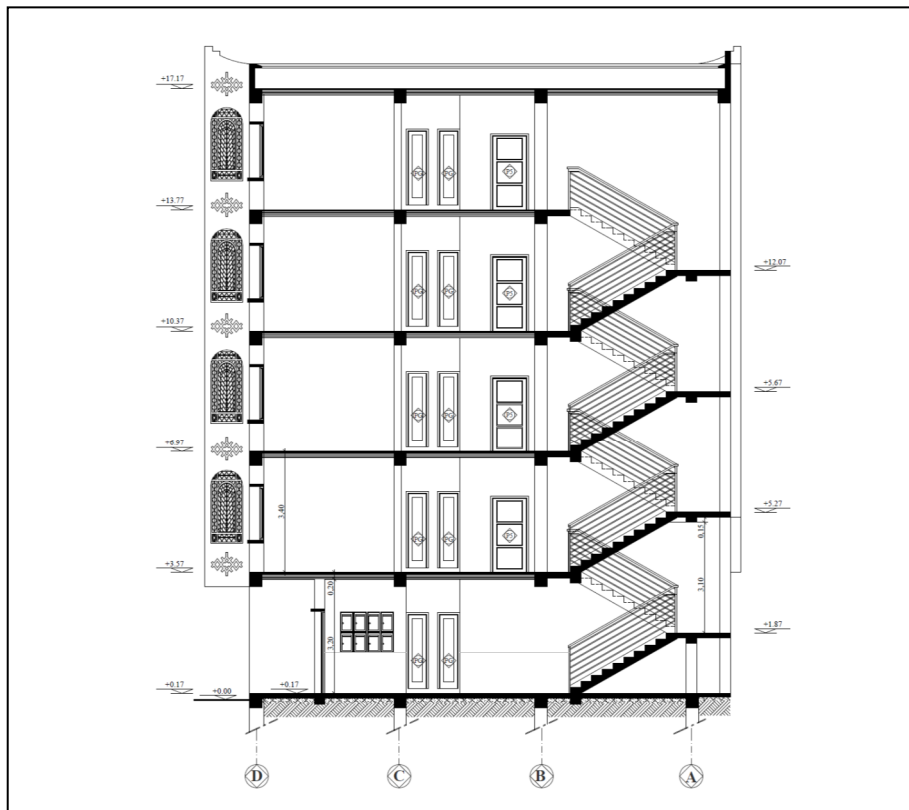
Figures II.3. Façade principale.

II.3. Les coupes.

En mode de représentation normal, les formes intérieures d'un objet simple peuvent être décrites à partir des traits interrompus courts ou pointillés, cependant la méthode devient vite complexe lorsque les contours intérieurs sont nombreux. Dans le cas des dessins d'ensemble, les tracés deviennent vite illisibles et l'identification des pièces impossible. Pour de tels cas, les vues en coupe, également appelées "coupes", permettent une meilleure définition et une compréhension plus aisée des formes intérieures ou des divers composants. Il existe plusieurs variantes de représentations répondant à des besoins différents.

La plupart utilise un plan de coupe imaginaire qui coupe l'objet en deux, la partie avant (celle en arrière du plan de coupe) est supprimée afin de pouvoir observer et dessiner les formes intérieures. Les hachures, tracées en traits fins, matérialisent le plan de coupe et mettent en évidence les contours intérieurs. Dans un dessin, une seule vue est concernée par une coupe, les autres vues restent en mode de représentation normal.

Lorsque le plan de coupe sectionne l'objet dans le sens de sa plus grande longueur, on parle de coupe longitudinale. Si la coupe est perpendiculaire à ce sens, la coupe est dite transversale.



Figures II.4. Coupe AA

II.4. Conclusion.

Le dessin technique est largement présenté dans la littérature. En effet, chaque partie est normalisée avec une utilisation bien précise comme les projections, les traits, les cotations, les hachures et les cartouches.

Le dessin technique approche la réalité au plus près. Sa qualité première est la précision, notamment s'il prélude à une mise en fabrication ou à une construction. Les proportions doivent donc être exactes, les perspectives fluides et le message lisible de tous.

De ce fait, il doit être aussi précis que possible et parfaitement conforme à la normalisation en vigueur, autant pour sa compréhension que pour son exécution. Les dessins techniques sont des documents non figés mais évolutifs qui changent avec la conception avec des améliorations ou simplifications, les matériaux, les fournisseurs et les utilisations.

Cette rigueur est prise en charge par des techniques de dessin qui ont évolué de façon extraordinaire au cours de ce siècle et, plus encore, au cours des dernières décennies grâce à l'apparition des outils numériques. Il s'agit du dessin assisté par ordinateur qui sera l'objet du chapitre prochain.

Chapitre III

Le dessin assisté par ordinateur

III.1. Introduction.

Autrefois, le dessin technique, se faisait sur une planche à dessin, à l'aide de la règle, du T, de l'équerre, du crayon et du compas. Il s'agit dans ce cas-là du dessin traditionnel, ou du dessin aux instruments.

Aujourd'hui, il arrive encore que dans les bureaux d'études où l'on retouche de cette manière d'anciens projets non encore numérisés (Charreb-Yssad, 2016).

Dés 1963, est apparu le logiciel Sketchpad, créé par Ivan Sutherland et qui est considéré comme le premier logiciel de DAO. Il comporte tous les concepts et outils considérés aujourd'hui comme basique : accrochage au point, bloc, copie, annulation.

Désormais, le Dessin Assisté par Ordinateur (D.A.O) domine largement ce domaine. La souris et le clavier ont remplacé le crayon et les autres instruments du dessinateur, mais l'objectif et le résultat sont identiques.

III.2. Définitions.

Le dessin assisté par ordinateur permet de produire des dessins techniques avec un logiciel informatique. C'est un outil d'aide à la production de dessins de communication, où la communication est prise ici dans le sens de la transmission d'informations sous forme de plans (Tourpe, 2004).

Les dessins produits sont le plus souvent réalisés en mode vectoriel. En effet, chaque élément d'un dessin est appelé entité, et chaque entité contient donc des propriétés de couleur, d'épaisseur, de calque, de type de ligne et d'autres propriétés encore.

III.3. Les objectifs du dessin assisté par ordinateur.

Le dessin assisté par ordinateur peut avoir différents objectifs. Il peut tout aussi bien servir à un électricien pour dessiner le schéma technique du circuit d'alimentation d'un édifice, qu'être utilisé par un graphiste. Il s'agit d'outils informatiques qui transposent dans le monde numérique des travaux auparavant réalisés d'un trait de crayon. La finalité n'est pas tant d'améliorer la qualité du dessin, mais de faciliter les modifications, le partage ou encore la sauvegarde. Le technicien de bureau d'études donne vie aux objets.

Grâce à la conception assistée par ordinateur, il les modélise en 3D, les assemble virtuellement et effectue une série de simulations. Avec le dessin assisté par ordinateur, il réalise plans et schémas détaillés qui seront transmis aux ateliers de fabrication.

Enfin, l'Architecte conçoit et dessine les plans sur lesquels sera basée la réalisation d'un édifice. Les domaines d'application sont ainsi nombreux. Quelques-uns sont présentés par la figure II.1.

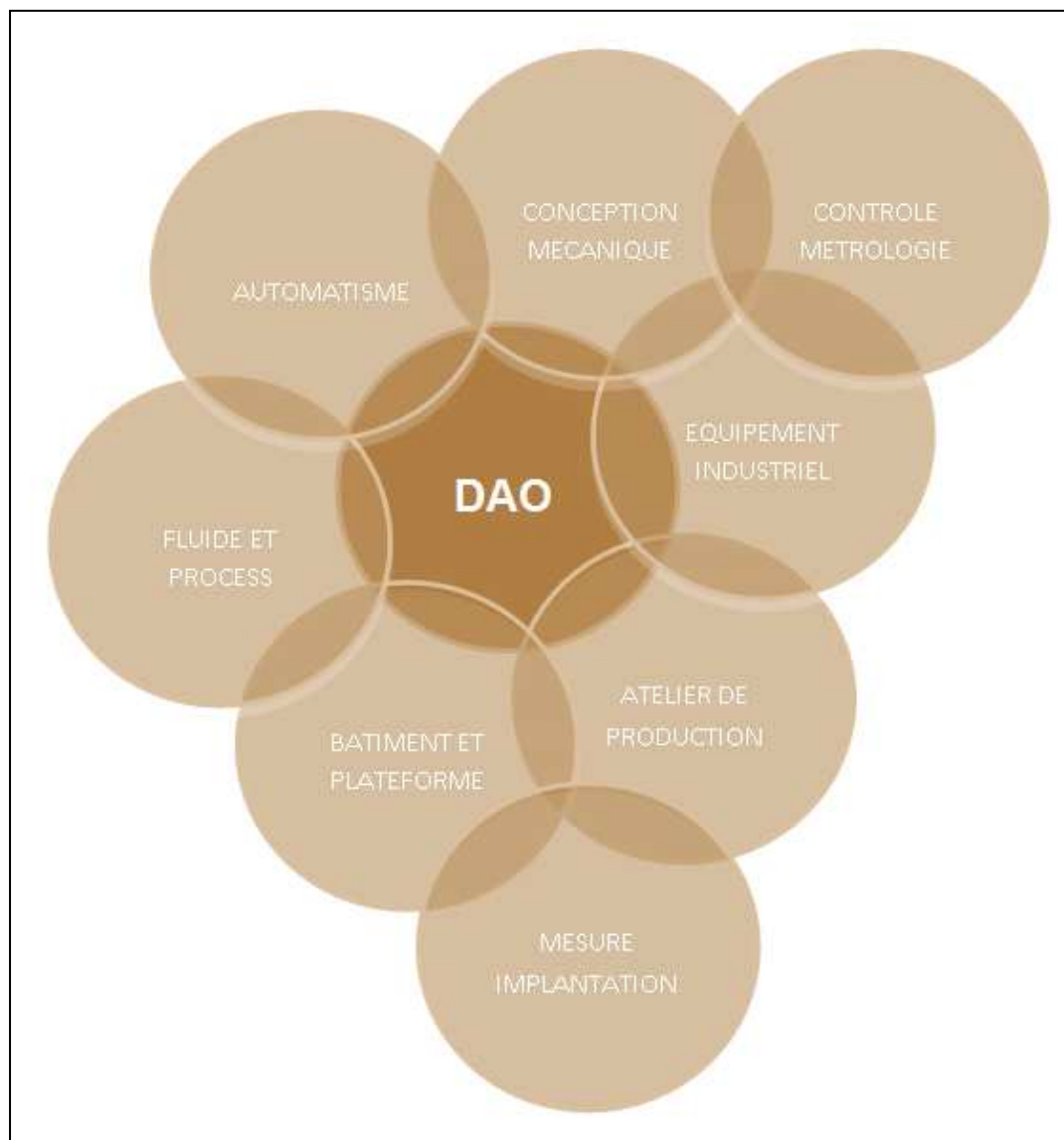


Figure III.1. Domaines d'applications du dessin assisté par ordinateur.

III.4. Les principes de base du dessin assisté par ordinateur

Pour réaliser un dessin, les logiciels de DAO utilisent et juxtaposent des éléments graphiques appelés primitives.

Ces primitives sont simplement les points, les segments de droite, les segments d'arc de cercle ou d'ellipse, les rectangles, les polygones les courbes, ou encore le texte.

Ces primitives sont représentées par des expressions algébriques qui permettent leur manipulation.

En plus du tracé des primitives qui permettent de construire un dessin complexe, le logiciel de DAO comporte d'autres fonctionnalités telles que :

- la modification d'échelle
- la possibilité de créer des calques,
- la cotation automatique
- la sortie des dessins sur imprimante ou traceur

III.5. Les avantages du dessin assisté par ordinateur.

En plus de simplifier la phase complexe de représentation du dessin, les systèmes de DAO offrent la possibilité de faciliter la mise en page et la production de plans comme le choix des échelles, des coupes, des vues et de leur disposition et des détails.

Une modification du projet dans sa phase de représentation est directement répercutée dans la phase d'impression des plans, contrairement au dessin à main-levée où tout serait à refaire.

Les avantages du dessin assisté par ordinateur sont intimement liées aux avantages de l'informatique. En effet, la grande capacité de stockage des informations permettant l'archivage des dessins et, par conséquent, la réutilisation future de tout ou partie d'un projet ainsi que le partage et le transfert des informations entre utilisateurs ou entre machines. Ceci est classique dans des projets de grande envergure, impliquant des intervenants de différentes disciplines qui peuvent, sans difficulté, utiliser les mêmes dessins.

Les logiciels de DAO permettent de tout dessiner en deux et/ou en trois dimensions, de « l'infiniment petit » à « l'infiniment grand ». Ils bénéficient de la puissance de calcul des processeurs, ce qui donne une facilité de travail très importante.

Par ailleurs, le dessin assisté par ordinateur offre une série d'outils facilitant la réalisation des phases fastidieuses telles que la copie multiple de mêmes éléments, la récupération d'éléments déjà dessinés dans un autre projet.

Dans un dessin très complet, même exagérément chargé d'informations, il est possible de ne visualiser temporairement qu'une partie grâce aux calques.

La modélisation impose souvent des allers et retours entre le plan, l'élévation et la coupe. Le modèle 3D les rassemble. Les systèmes de DAO ont l'avantage de pouvoir manipuler aisément ces objets. Le passage instantané d'une représentation volumique à une représentation plane est possible, par la définition d'un plan de coupe ou de vecteurs de projections. A l'inverse, un plan tracé en deux dimensions est une excellente base de travail pour représenter plus facilement le volume correspondant en trois dimensions (Tourpe, 2004).

III.6. Les logiciels du dessin assisté par ordinateur.

Il existe autant de logiciels de DAO que de métiers utilisant le dessin. Le mécanicien, l'architecte, mais aussi l'électricien et le géomètre disposent aujourd'hui d'outils facilitant la création d'un plan, d'un schéma, avec des commandes orientées métiers, des bases de données adaptées, comme par exemple : Archicad, Autocad, Sketchup, 3D Studio Max, Maya, Revit, Inventor, MicroStation, SolidWorks.

Les primitives de dessin de ces logiciels obéissent aux mêmes principes du dessin vectoriel. Le dessin se fait soit par des commandes, des options du menus, des icônes ou bien par programmation. La figure III.2 montre un exemple de commandes vectorielles sous Autodesk-Autocad.

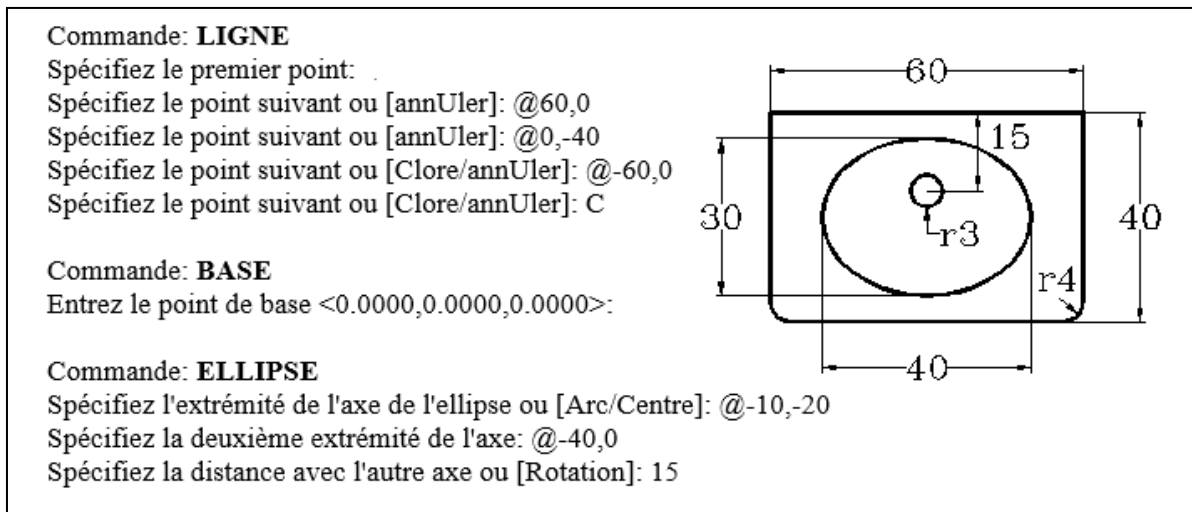


Figure III.2. Exemples de commandes sous Autodesk-Autocad

III.5. Conclusion.

De la préhistoire à nos jours, toutes les générations d'hommes ont dessiné. C'est par le dessin qu'un architecte impose son art ou qu'un mécanicien fait naître de nouvelles machines avec comme base, le dessin technique.

L'avènement du dessin assisté par ordinateur a démocratisé l'accès aux méthodes du dessin technique. Ainsi, les étudiants en architecture ne sont plus obligés d'acheter des tables et du matériel de dessin coûteux et bien souvent inaccessibles aux plus modestes.

Malgré sa puissance et ses nombreux avantages, le dessin assisté par ordinateur ne permet pas de prendre en charge correctement des projets complexes. Le dessin paramétrique apporte des solutions novatrices à ces situations en profitant de la puissance de calcul des ordinateurs qui ne cesse de progresser. Ainsi, le prochain chapitre sera consacré aux principes de base du dessin paramétrique.

Chapitre IV

Les principes de base du dessin paramétrique

IV.1. Introduction.

Le dessin modélisation paramétrique permet aux dessinateurs d'adapter les paramètres d'un dessin géométrique à chaque modification qui est réalisée. Les dimensions peuvent être des variables, c'est-à-dire on peut donner une cote à un élément géométrique qui dépend d'une autre cote. Par conséquent, toute modification subite à la première cote engendrera automatiquement une modification sur la deuxième cote selon ce qu'on a défini comme variable.

Il est possible de procéder ainsi pour l'ensemble des paramètres du dessin qui pourraient varier.

IV.2. Du dessin assisté par ordinateur au dessin paramétrique.

Le dessin assisté par ordinateur classique prend en charge les formes des plus simples aux plus complexes en 2 dimensions. La figure IV.1 en montre un exemple.

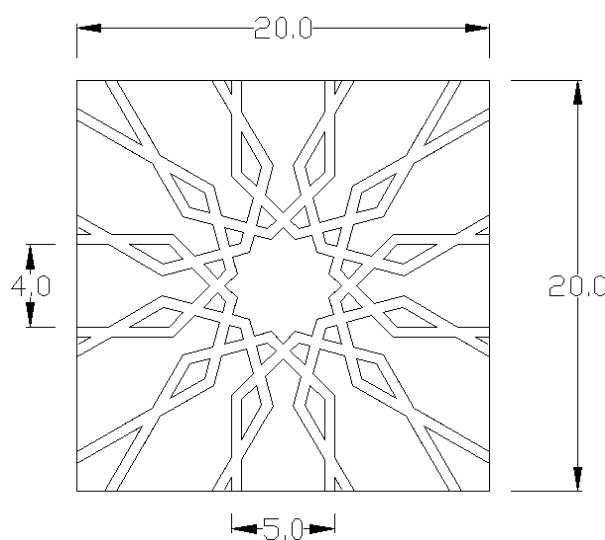


Figure IV.1. Exemple d'un dessin en 2 dimensions

Il prend également en charge les formes en 3 dimensions. Soit en extrudant des formes préalablement réalisées en 2 dimensions comme le montre la figure IV.2, soit en exploitant des modèles préexistants des formes usuelles en 3D.

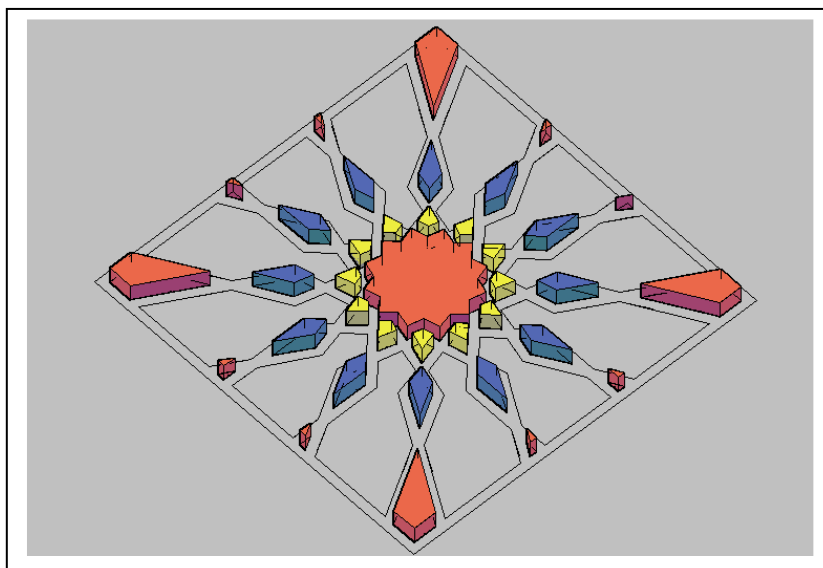


Figure IV.2. Un dessin en 3 dimensions extrudé à partir d'un autre dessin en 2 dimensions.

Ces dessins qui paraissent de premier abord complexes, sont en réalité assez faciles à réaliser avec une répétition de certaines formes de base. Ainsi, il est possible d'obtenir des variantes différentes en faisant varier le nombre de répétition, les angles et les longueurs des formes de base. La figure IV.3 montre une variante du précédent dessin représenté par la figure IV.1.

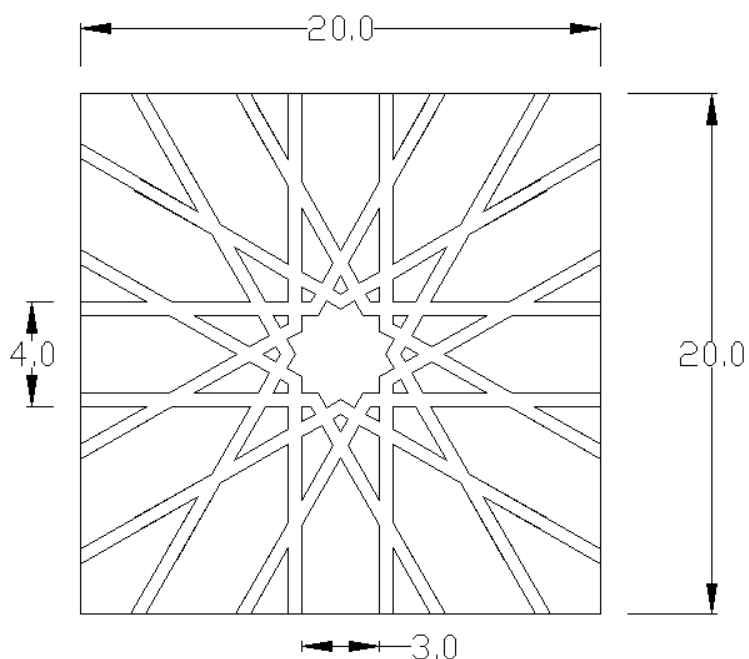


Figure IV.3. Variante en 2 dimensions de la figure IV.1

Cette variation des valeurs des paramètres que sont le nombre de répétitions, l'angle entre les formes de base et leurs longueurs constitue l'essence même du dessin paramétrique qui permet d'obtenir de nouvelles figures à chaque fois que les valeurs des paramètres varient.

IV.3. Le rôle de la programmation dans le dessin paramétrique.

La programmation joue un rôle primordial pour la gestion et la manipulation des objets d'un dessin. En effet, les lignes, les arcs, le texte et les cotes sont des objets.

Les paramètres de style comme les types de ligne et les styles de cotes sont des objets. Les éléments de structure comme les calques, les groupes et les blocs sont aussi des objets. Même les éléments d'affichage graphique comme les vues et les fenêtres sont des objets. La figure IV montre un algorithme qui fait varier la forme de base qui est un segment selon un angle de 40° pendant 8 fois pour aboutir au dessin d'un octogone.

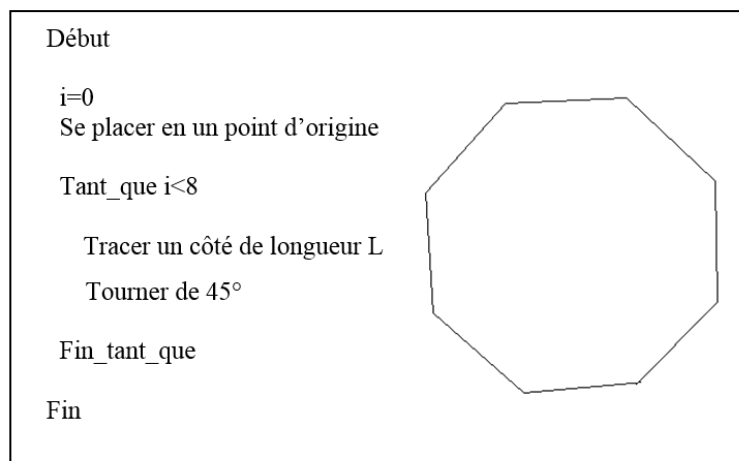


Figure IV.4. Exemple d'un algorithme de dessin et le résultat obtenu

Il est possible de créer des formulaires afin de permettre à l'utilisateur de saisir de nouvelles valeurs des paramètres, permettant d'obtenir un nouveau dessin à chaque tentative. La figure IV.5 montre un exemple de formulaire permettant de dessiner des figures en faisant varier l'angle de rotation, le nombre et la longueur des segments.

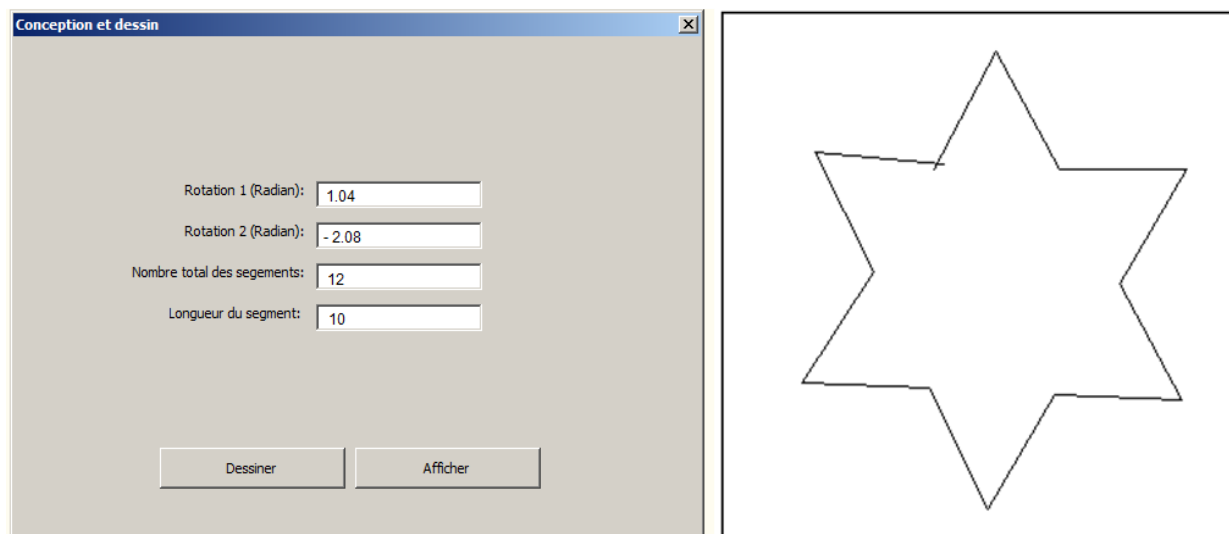


Figure IV.5. Formulaire permettant la saisie des valeurs des paramètres et l'exécution du programme de dessin.

Ainsi, le dessin paramétrique permet de répondre en un temps record à de nouveaux besoins qui pourraient apparaître en faisant varier les valeurs des paramètres essentiels. Certes, c'est une démarche plus complexe mais les performances sont sans commune mesure avec le dessin classique.

IV.4. Le dessin aléatoire.

L'introduction du hasard dans le dessin permet d'explorer des pistes insoupçonnées dans la création d'un projet. En effet, le dessin aléatoire offre des résultats surprenants à chaque tentative. La figure IV.6 montre le formulaire et le résultat d'un dessin aléatoire. Ainsi, un polyline de 200 000 segments est obtenu avec un résultat différent à chaque tentative.

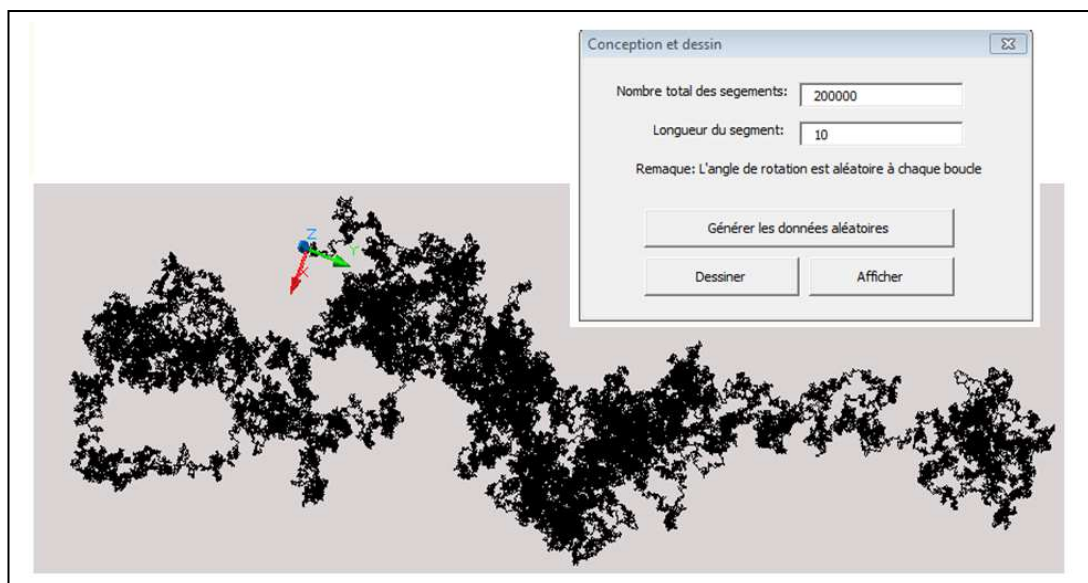


Figure IV.5. Formulaire et résultat d'un dessin aléatoire.

Il est possible d'intégrer des contraintes dans le programme du dessin aléatoire afin d'aboutir au résultat désiré grâce à un hasard contraint.

Par exemple, les contraintes peuvent concerner la délimitation du dessin qui ne pourra pas s'étendre au-delà de limites prédéfinies comme le montre la figure IV.6. Il s'agit d'un dessin de 2000 segments avec des angles de rotations aléatoires entre chaque paire de segments contigus. Cependant, le dessin des segments est contraint par des limites.

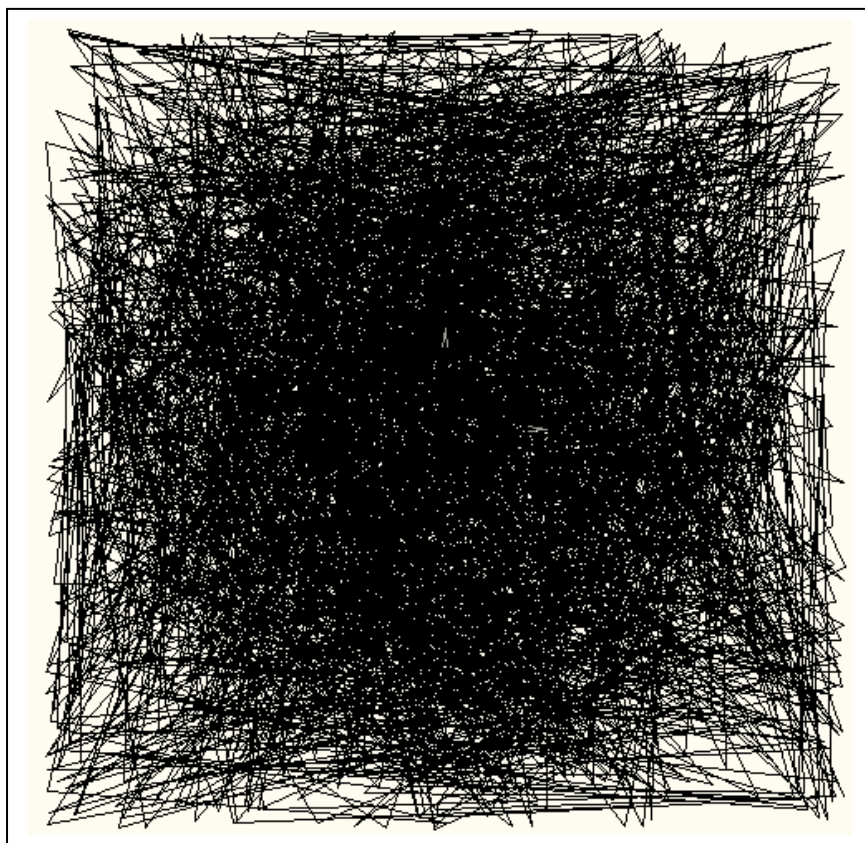


Figure IV.5. 2000 segments dessinés aléatoirement mais contraints par des limites.

Des hachures aléatoires pourront être ajoutées afin de matérialiser des ouvertures générées par le hasard. La figure IV.7 en montre un exemple.

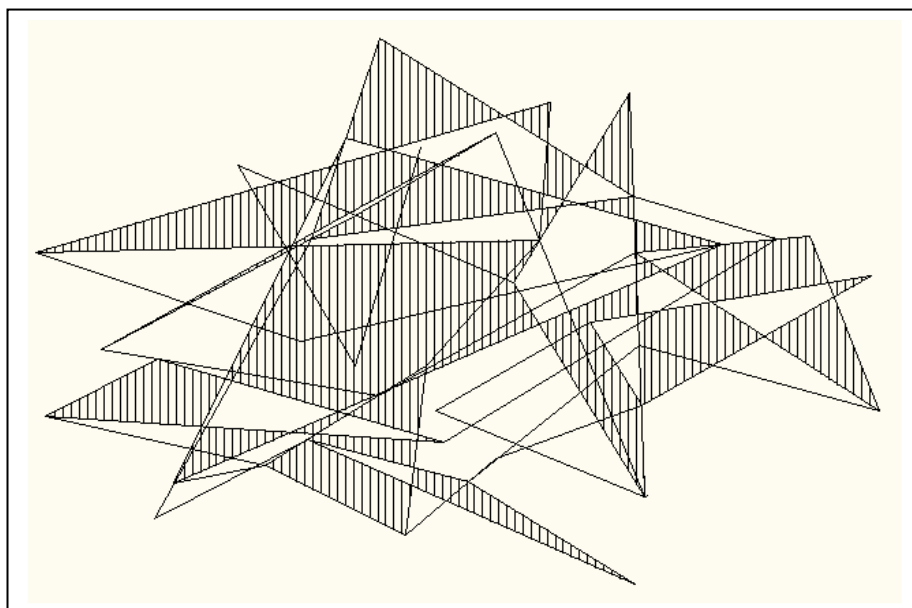


Figure IV.7. Segments et hachures générées aléatoirement dans des limites prédéfinies

Les possibilités du dessin paramétriques basé sur le hasard sont innombrables. La puissance de calcul de l'ordinateur permet de générer aléatoirement des milliers de variantes et d'en éliminer automatiquement celles qui ne respectent pas les contraintes prédéfinies. Ainsi, le concepteur aura plus de facilité à explorer tous les recoins des théories qu'il pourrait développer à condition qu'il sache exploiter intelligemment ce formidable outil d'aide à la création.

IV.5. Conclusion.

Le dessin paramétrique est fondé sur des objets dont la géométrie peut être pilotée par des paramètres et des formules logiques, et des objets capables de véhiculer de l'information et d'interagir avec leur contexte d'une manière dynamique.

Le dessin paramétrique peut représenter alors un support intéressant aux architectes en étant un outil d'aide à leurs créativité. Cependant, il est nécessaire d'acquérir au préalable les bases de la programmation et du raisonnement algorithmique. Le prochain chapitre y sera consacré.

Chapitre V

Notions de programmation.

V.1. Introduction.

Un programme informatique est une suite d'instructions interprétées/compilées puis exécutées par un ordinateur.

Toutefois, l'ordinateur ne sait manipuler que du **binaire**, c'est-à-dire une succession de 0 et de 1. Il est donc nécessaire d'utiliser un **langage de programmation** pour écrire de façon lisible les instructions à exécuter par l'ordinateur, c'est-à-dire avec des instructions compréhensibles par l'humain car proches de son langage.

Ainsi, ces programmes sont traduits en langage machine (en binaire) par un compilateur.

D'une façon générale, le programme est un simple fichier texte (écrit avec un traitement de texte ou un éditeur de texte), que l'on appelle **fichier source**.

Le fichier source contient les lignes de programmes et doit être compilé afin de générer un fichier exécutable.



Cette nouvelle version du programme source compilé, appelée programme exécutable, est alors prête à fonctionner.

En interdisant l'accès aux instructions codées en langage machine, un programme exécutable conserve en son sein la confidentialité des algorithmes mis au point par le programmeur. Ainsi, une application de traitement de texte constitue un programme exécutable : l'utilisateur n'a pas besoin de modifier le programme pour l'exécuter, et quand bien même il le désirerait, il n'en aurait pas la possibilité.

V.2. Le développement d'un programme.

La réalisation ou développement d'un programme passe par une phase conceptuelle indispensable durant laquelle le programmeur, appelé encore développeur, écrit ce programme au moyen d'instructions appartenant au langage de programmation choisi. Mais le seul fait de connaître un langage n'implique pas forcément la création d'un bon programme. Il est fondamental de concevoir d'abord l'algorithme qui résolve le problème pour ensuite le traduire en langage de programmation. D'autres connaissances supplémentaires peuvent, en effet, être nécessaires, telles que la conception de l'interface utilisateur, la connaissance des réseaux, des serveurs de données, etc.

Le développement d'un programme ou d'un logiciel s'inscrit dans un cycle qui se décompose en une succession d'étapes :

- élaboration du cahier des charges
- analyse du système et du problème à résoudre,
- programmation,
- validation et tests
- installation et déploiement,
- maintenance (maintien en condition opérationnelle).

La programmation adoptée dépend souvent du **type d'application** que l'on souhaite développer, ce choix prenant en compte, suivant les cas, des critères de rapidité, de facilité de traitement du graphisme ou de calcul, etc.

V.3. Les langages de programmation.

- FORTRAN (*FORmula TRANslation*): Premier langage de programmation de haut niveau, entre 1954 et 1958. Utilisé actuellement pour la simulation et la modélisation mathématique.
- Pascal : Langage utilisé dans un but pédagogique et académique

- C / C++ : Très puissant, permet de gérer une grande partie des ressources de la machine
- Prolog (*PRO*gramming in *LOGic*) : Conçu à l'origine pour la programmation logique, utilisé dans le domaine de l'intelligence artificielle.
- SQL *Structured Query Language* : utilisé dans les systèmes de gestion de bases de données, permettant notamment de définir les structures des données et d'interroger les bases de données existantes.
- Java : Utilisé pour développer des applications pour Internet. Le langage Java est portable, c'est-à-dire qu'il peut être interprété par de nombreuses plates-formes
- Visual-Basic, Auto-Lisp, Ada, ...

V.4. La notion d'algorithme.

V.4.1. Historique.

Le mot **algorithme** est dérivé d'une contraction et d'une dérivation du nom du célèbre mathématicien arabe Al Khawarizmi qui publia deux livres importants : l'un sur l'arithmétique et l'autre sur « l'action de faire passer et d'agencer les parties d'un tout » (titre original : *Kitab al-jabr wal muqabala*). Trois siècles plus tard, le livre, traduit en latin, porta le nom *Algorismus*.

V.4.2. Définition.

La première définition du mot algorithme, dans son sens actuel, a été donnée par le mathématicien russe Markov : « Tout ensemble de règles précises qui définit un procédé de calcul destiné à obtenir un résultat déterminé à partir de certaines données initiales.»

Les algorithmes sont constitués par un ensemble de règles précises et compréhensibles par tous. Ils s'appliquent à des données qui peuvent changer et donnent les résultats en fonction des données initiales.

V.4.3. Liste d'instructions.

Un algorithme se présente comme une **liste d'instructions** de nature mathématique ou logique pouvant être programmées ; il peut se traduire schématiquement par un organigramme.

Les algorithmes informatiques offrent des performances bien supérieures aux méthodes de résolutions manuelles et peuvent être appliqués à des problèmes de taille importante.

Un algorithme informatique doit répondre à des règles :

- il doit être fini et se terminer après un nombre déterminé d'opérations,
- chaque instruction doit être définie sans ambiguïté,
- le fonctionnement de l'algorithme est déterministe (il donne les mêmes résultats pour les mêmes données),
- toutes les opérations doivent pouvoir être effectuées par une personne en utilisant des moyens manuels (papier et stylo).

Il n'existe pas de méthode pour concevoir un algorithme. Sa réalisation est un acte **créatif**.
Un même problème peut être résolu au moyen de plusieurs algorithmes.

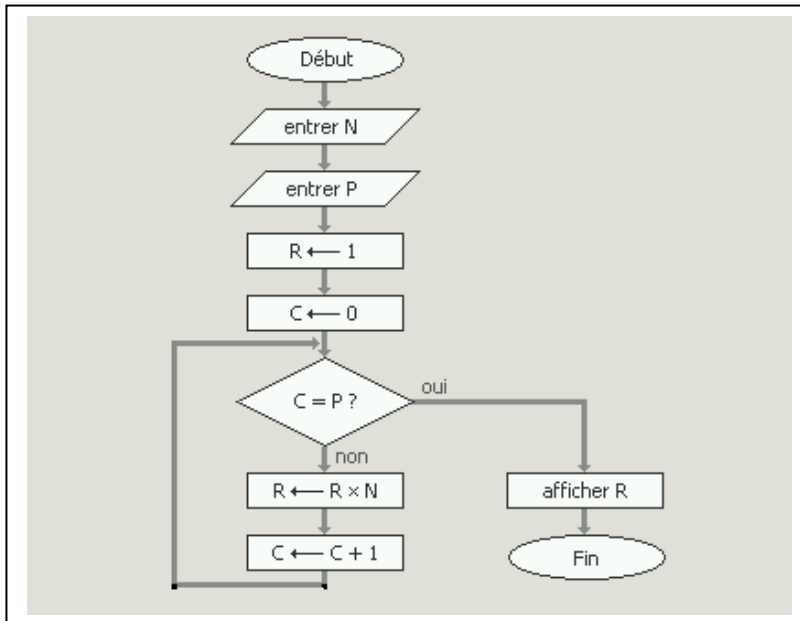
V.4.4. Les caractéristiques d'un algorithme.

L'algorithme est un moyen pour le programmeur de présenter son approche du problème à d'autres personnes. En effet, un algorithme est l'énoncé dans un langage bien défini d'une suite d'opérations permettant de répondre au problème. Un algorithme doit donc être :

- **lisible**: l'algorithme doit être compréhensible même par un non-informaticien
- **de haut niveau**: l'algorithme doit pouvoir être traduit en n'importe quel langage de programmation, il ne doit donc pas faire appel à des notions techniques relatives à un programme particulier ou bien à un système d'exploitation donné
- **précis**: chaque élément de l'algorithme ne doit pas porter à confusion, il est donc important de lever toute ambiguïté
- **concis**: un algorithme ne doit pas dépasser une page. Si c'est le cas, il faut décomposer le problème en plusieurs sous-problèmes
- **structuré**: un algorithme doit être composé de différentes parties facilement identifiables

L'Algorithme qui calcule N^P peut être soit sous forme d'un **organigramme** :

soit sous forme d'un programme en **pseudo-langage**



```

Début
Lire N
Lire P

R=1
C=0

Tant que C<>P
    R=R x N
    C=C+1
Fin_tant_que

Ecrire R
Fin
    
```

Soit sous forme d'un programme écrit selon le lexique, la syntaxe et la sémantique d'un langage de programmation (Visual Basic).

```

Sub Exercice01()
Dim N As Integer
Dim P As Integer
Dim C As Integer
Dim R As Integer

N = InputBox("Donner la valeur de N", "LMD_L2")
P = InputBox("Donner la valeur de P", "LMD_L2")

R = 1
C = 0

While C <> P
    R = R * N
    C = C + 1
Wend

MsgBox "Le résultat est : " & R, vbInformation, "LMD_L2"
End sub
    
```

Déclaration des variables

Saisie des données par l'utilisateur

Boucle

Affichage du résultat

V.5.Comment programmer ?

Après avoir conçu l'algorithme, il s'agit de le traduire selon le lexique, la syntaxe et la sémantique du langage de programmation choisi.

a- La notion de variable.

Dans la plupart des langages, on travaille généralement sur des variables, c'est-à-dire que l'on associe un nom à un contenu. On pourra ainsi appeler une variable "Archi" et y stocker le chiffre 7.

Les noms de variables admettent généralement une longueur maximale (qui dépend du langage) et un jeu de caractères réduit.

b- Les mots réservés.

Dans la plupart des langages, il existe des mots que l'on ne peut pas attribuer aux noms de variables, on les appelle **mots réservés** (en anglais *reserved words*). Par exemple : while, if, then, else pour le langage VBA.

c- Les constantes.

Les constantes sont des données dont la valeur ne peut être modifiée. On les définit généralement en début de programme. La valeur que la constante contient peut être de tout type, suivant ce que le langage autorise.

d- Les commentaires.

Il est généralement bon de pouvoir ajouter dans un programme des lignes de texte qui ne seront pas prises en compte par le compilateur. Ces lignes de textes sont généralement précédées (ou encadrées) par des instructions spéciales qui signaleront au compilateur de les ignorer.

Les commentaires servent à clarifier un programme en donnant des explications. Ils serviront si jamais une autre personne essaie de comprendre le fonctionnement du programme en lisant le fichier source, ou bien à la personne qui l'a créé si jamais il relit le fichier source quelques années après l'avoir écrit...

e- Les instructions.

L'instruction est l'élément clé de l'ordinateur car c'est elle qui permet de spécifier au processeur l'action à effectuer. Les instructions à effectuer sont indiquées dans le fichier source et l'ordinateur passe d'une instruction à une autre en suivant les instructions indiquées de haut en bas (car la lecture d'un fichier se fait de haut en bas).

Les instructions d'un langage ont la propriété d'être non ambiguës, c'est-à-dire que pour chacune d'elles, l'ordinateur effectue une seule action et si l'instruction contient des erreurs (omission de paramètres, etc.), elle est incompréhensible pour la machine. Leur écriture est soumise au respect des règles de syntaxe qui constituent la grammaire du langage. De plus, il faut rappeler que les programmes sont lus séquentiellement de haut en bas et que les instructions sont exécutées les unes après les autres, sauf indications contraires.

➤ Instructions d'affectation

En langage VBA, l'instruction qui attribue la valeur 20 à la variable AGE s'écrit :

$$AGE = 20$$

où « = » est l'opérateur d'affectation. Il signifie que l'expression du membre de droite (20) est évaluée, et que sa valeur est affectée (ou assignée) à la variable spécifiée dans le membre de gauche AGE. Le membre de droite peut être une constante, une variable, une expression arithmétique, logique ou relationnelle, ou encore un sous-programme renvoyant une valeur.

➤ **Instructions conditionnelles.**

L'exécution d'une instruction ou d'un bloc d'instructions conditionnelles est déterminée par la réalisation d'une condition et donc le renvoi à des traitements différents. La structure d'une instruction conditionnelle associée à la description d'une alternative est identique dans tous les langages de programmation et a la forme suivante :

SI *condition* ALORS

instruction 1

instruction 2

...

instruction X

SINON

instruction 1

instruction 2

...

instruction X

FIN SI

Si la condition est vraie, le bloc sous contrôle (ALORS) est **exécuté** ; sinon il est **ignoré** et le bloc suivant (SINON) est **exécuté**.

➤ **Instructions répétitives.**

Un programme itératif est un programme exécutant plusieurs fois le même travail. Le programme itératif, appelé également boucle, itération ou répétition, permet de spécifier le nombre de traitements à l'aide d'une condition d'arrêt, qui n'est pas fixée au démarrage mais évaluée lors de l'exécution. La notion d'itération est fondamentale pour la programmation d'algorithmes.

Risque d'erreur : Programme sans fin.

L'itération est un type de programme dangereux, car elle peut ne pas s'arrêter. En effet, bien qu'une condition d'arrêt soit prévue dans l'algorithme, elle peut ne jamais être atteinte.

L'itération peut prendre trois formes : la première consiste à répéter un ensemble d'instructions jusqu'à ce qu'une certaine condition soit réalisée. Prenons l'algorithme suivant.

```
RÉPÉTER  
LIRE Nombre  
AFFICHER Nombre  
JUSQU'À Nombre = 0
```

Les mots *RÉPÉTER* et *JUSQU'À* encadrent le bloc d'instructions « *AFFICHER Nombre* ». Cela signifie que ces instructions doivent être répétées autant de fois qu'il est nécessaire et ceci jusqu'à ce que la variable « *Nombre* » prenne la valeur 0 (zéro). Dans cet exemple, le nombre de répétitions du bloc d'instructions n'est pas explicitement indiqué ; il dépend des données saisies par l'opérateur. De plus, la condition « *nombre = 0* » n'est examinée qu'après l'exécution du bloc d'instructions.

L'instruction *TANT QUE* associée à un compteur permet la répétition d'un bloc d'instructions selon un nombre connu de fois. Par exemple : réaliser la moyenne de 360 notes, etc. Prenons l'algorithme suivant.

```
TANT QUE Nombre < Nombre_étudiants  
calculer_moyenne  
Nombre = Nombre + 1  
FIN TANT QUE
```


Le même résultat est obtenu avec l'instruction POUR... DE... À... Par exemple :

```
POUR Nombre de 1 à nombre_étudiants  
calculer_moyenne  
FIN POUR
```

V.6. Diffusion des logiciels.

Les logiciels, suivant leurs tailles, peuvent être développés par une personne seule, une petite équipe, ou un ensemble d'équipes coordonnées. Le développement de grands logiciels par de grandes équipes pose de grands problèmes de coordination, en raison de la quantité importante d'informations à communiquer entre les intervenants.

Il existe plusieurs types de logiciels, parmi lesquels :

- Les logiciels commerciaux standard : développés par les éditeurs de logiciels (Microsoft, Autodesk, Adobe, Oracle, Borland, ...) et vendus essentiellement par des revendeurs avec leurs licences d'utilisation.
- Les logiciels du domaine public à distribution gratuite : comprennent le *freeware*, à usage libre et sans droits à payer, et le *shareware*, dont l'utilisateur régulier devra payer une somme modique à son créateur.

V.7. Conclusion.

La programmation informatique est utile dans un grand nombre de disciplines. En effet, l'omniprésence du numérique dans notre quotidien nécessite une exploitation, un traitement et une analyse fine des informations qui nous entourent.

La programmation sert à analyser les problèmes et à les résoudre à l'aide d'algorithmes, puis à traduire ceux-ci dans l'un des nombreux langages de programmation existants. Si ces problèmes sont de nature architecturale complexe alors on pourra faire intervenir la programmation et les techniques du dessin paramétriques qui pourront y apporter des solutions novatrices.

En effet, la programmation est intimement liée au dessin paramétrique. Elle sert à manipuler les données, à les traiter et à les transformer en dessins. Cependant, si un projet comporte des données massives alors le problème du stockage de ces données va se poser. Les structures les plus aptes à gérer une grande quantité d'information sont les bases de données dont les caractéristiques seront présentées lors du prochain chapitre.

Chapitre VI

Les bases de données

VI.1. Introduction.

La bonne compréhension des bases de données est fort utile pour utiliser le dessin paramétrique en manipulant des données massives. De plus, le dessin paramétrique lié à une base de données épouse parfaitement les impératifs de la démarche BIM (Building Information Modeling) dont la puissance de traitement de l'information et la finesse des analyses découlent principalement des systèmes de gestion de bases de données qui y sont intégrés.

VI.2. Définition d'une base de données

Une base de données (son abréviation est BD, en anglais DB, *database*) est une organisation particulière d'une masse d'informations ou d'observations. C'est une entité dans laquelle il est possible de stocker des données de façon structurée et avec le moins de redondance (répétition) possible.

Son but est donc de simplifier l'accès aux données et d'économiser l'espace occupé par les fichiers informatiques en réduisant la redondance d'informations.

Les données doivent pouvoir être utilisées par des programmes, par des utilisateurs différents. Ainsi, la notion de base de données est généralement couplée à celle de réseau, afin de pouvoir mettre en commun ces informations, d'où le nom de **base**. On parle généralement de système d'information pour désigner toute la structure regroupant les moyens mis en place pour pouvoir partager des données selon une architecture clients-serveur ou trois-tiers comme le montre la figure VI.1.

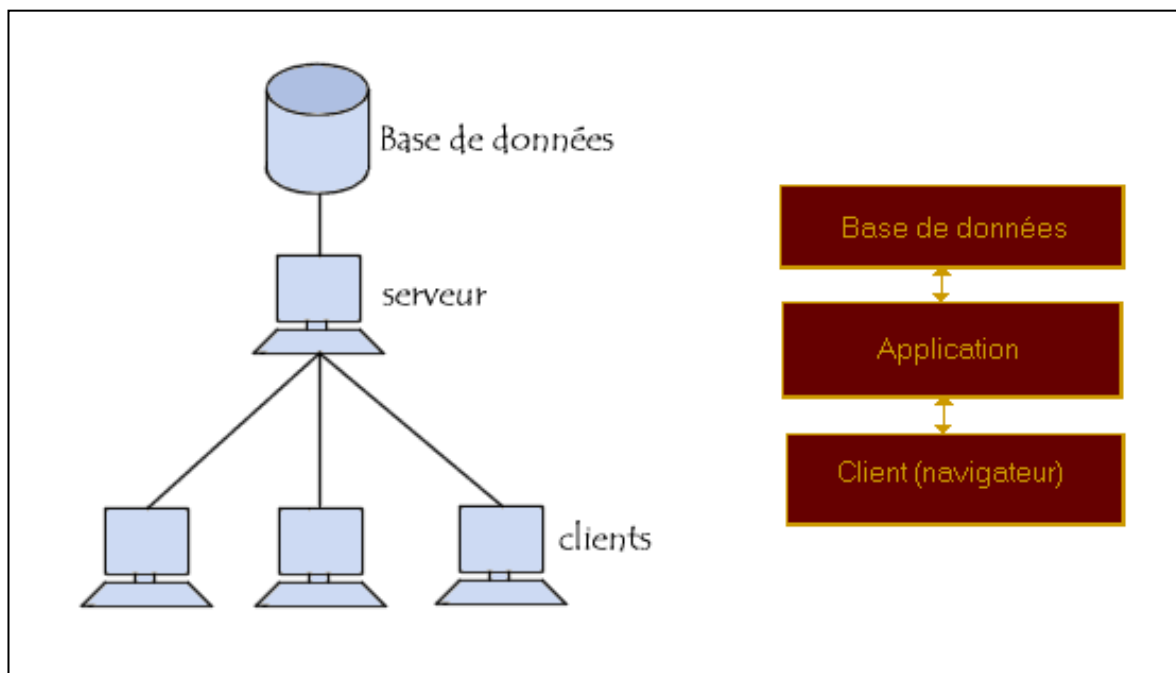


Figure VI.1 : Architecture clients-serveur et architecture trois tiers des bases de données.

VI.3. Utilité d'une base de données.

Comment gérer (mémoriser et traiter) un ensemble d'étudiants (n°, nom, prénom, groupe, filière...)?

Première solution : L'utilisation d'un tableur (Excel par exemple)

Mais comment traiter la requête suivante ?

- Quelle est la moyenne des Tests des redoublants, qui ont plus de 3 absences et qui habitent Oran ?

Un tableur ne répond pas totalement à ce besoin surtout si le nombre d'étudiant est important.

➤ **Solution**

- Séparation entre les données et le traitement
- Utilisation d'un Système de Gestion de Base de Données (SGBD)

Une base de données permet de mettre des données à la disposition d'utilisateurs pour une **consultation**, une **saisie** ou bien une **mise à jour**, tout en s'assurant des droits accordés à ces derniers. Cela est d'autant plus utile que les données informatiques sont de plus en plus nombreuses.

Une base de données peut être locale, c'est-à-dire utilisable sur une machine par un utilisateur, ou bien répartie, c'est-à-dire que les informations sont stockées sur des machines distantes et accessibles par réseau.

L'avantage majeur de l'utilisation de bases de données est la possibilité de pouvoir être accédées par plusieurs utilisateurs simultanément.

Un carnet d'adresses peut constituer une base de données assez simple à concevoir. A l'opposé, le responsable d'un grand magasin peut rassembler le fichier du personnel, le fichier clients, la liste des produits en stock et les ventes par secteur et par vendeur dans une base de données complexe!

Le bon fonctionnement d'une base de données dépend avant tout de son organisation et c'est là où réside la principale difficulté. Il faut commencer par réfléchir à **la structure de l'information** avant de faire quoi que ce soit! Immédiatement après, il faudra bien cibler les besoins de l'utilisateur. Le choix du logiciel (MS-Access, Oracle, ...) ne vient qu'après et dépendra de la complexité et du volume des données.

VI.4. La gestion des bases de données.

Afin de pouvoir contrôler les données ainsi que les utilisateurs, le besoin d'un système de gestion s'est vite fait ressentir. La gestion de la base de données se fait grâce à un système appelé **SGBD** (système de gestion de bases de données) ou en anglais DBMS (Database management system).

Le SGBD est un ensemble de services (applications logicielles) permettant de gérer les bases de données, c'est-à-dire :

- permettre l'accès aux données de façon simple
- autoriser un accès aux informations à de multiples utilisateurs
- manipuler les données présentes dans la base de données (insertion, suppression, modification)

Le SGBD peut se décomposer en trois sous-systèmes (**Figure VI.2**) :

- le système de gestion de fichiers : il permet le stockage des informations sur un support physique
- le SGBD interne : il gère l'ordonnancement des informations
- le SGBD externe : il représente l'interface avec l'utilisateur

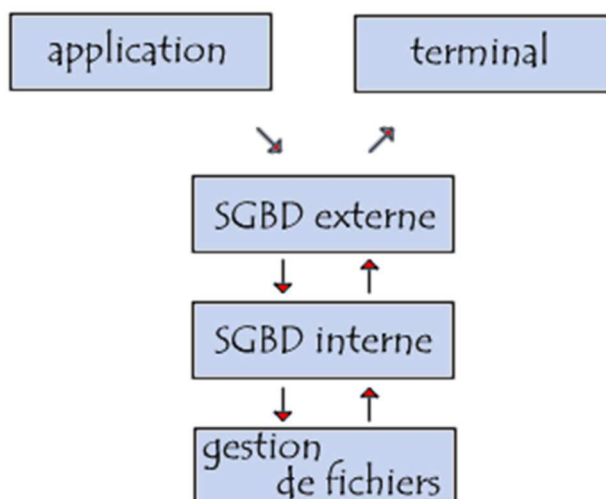


Figure VI.2. Les composants d'un système de gestion des bases de données.

VI.5. Les principaux SGBD.

Les principaux systèmes de gestion de bases de données sont les suivants :

- Borland Paradox
- IBM DB2
- Ingres
- Interbase
- Microsoft SQL server
- Microsoft Access
- Microsoft FoxPro
- Oracle
- Sybase
- MySQL

VI.6. Les caractéristiques d'un SGBD.

L'architecture à trois niveaux permet d'avoir une indépendance entre les données et les traitements. D'une manière générale un SGBD doit avoir les caractéristiques suivantes :

- **Indépendance physique** : le niveau physique peut être modifié indépendamment du niveau conceptuel. Cela signifie que tous les aspects matériels de la base de données n'apparaissent pas pour l'utilisateur, il s'agit simplement d'une structure transparente de représentation des informations.
- **Indépendance logique** : le niveau conceptuel doit pouvoir être modifié sans remettre en cause le niveau physique, c'est-à-dire que l'administrateur de la base doit pouvoir la faire évoluer sans que cela gêne les utilisateurs.
- **Manipulabilité** : des personnes ne connaissant pas la base de données doivent être capables de décrire leur requête sans faire référence à des éléments techniques de la base de données
- **Rapidité des accès** : le système doit pouvoir fournir les réponses aux requêtes, le plus rapidement possible, cela implique des algorithmes de recherche rapides.

- **Administration centralisée** : le SGBD doit permettre à l'administrateur de pouvoir manipuler les données, insérer des éléments, vérifier son intégrité de façon centralisée.
- **Limitation de la redondance** : le SGBD doit pouvoir éviter dans la mesure du possible des informations redondantes, afin d'éviter d'une part un gaspillage d'espace mémoire mais aussi des erreurs.
- **Vérification de l'intégrité** : les données doivent être cohérentes entre elles, de plus lorsque des éléments font référence à d'autres, ces derniers doivent être présents.
- **Partageabilité des données** : le SGBD doit permettre l'accès simultané à la base de données par plusieurs utilisateurs.
- **Sécurité des données** : le SGBD doit présenter des mécanismes permettant de gérer les droits d'accès aux données selon les utilisateurs.

VI.7. Les modèles de base de données.

Il existe plusieurs modèles de bases de données : le modèle hiérarchique, le modèle réseau, le modèle relationnel, ...

A la fin des années 90 les bases relationnelles sont les bases de données les plus répandues (environ trois quarts des bases de données).

Le modèle relationnel (SGBDR, *Système de gestion de bases de données relationnelles*) : les données sont enregistrées dans des tableaux à deux dimensions (lignes et colonnes). La manipulation de ces données se fait selon la théorie mathématique des relations.

VI.8. Exemple de base de données

Si le tableur (Excel par exemple) fonctionne par feuilles de données, sur lesquelles on trouve des tableaux, des calculs, des graphiques, ..., les logiciels permettant de travailler sur des bases de données (Prenons l'exemple d'une base de données **MS-Access**), eux, dissèquent chacune de ces étapes.

Ainsi, au moins quatre parties y sont bien distinctes, En effet, vous retrouvez :

- des tables
- des requêtes
- des formulaires
- des états
- des macros, des modules.

Quelle est la démarche à suivre alors ? Déjà, oublier la manière de travailler sur les tableurs tels qu'Excel (les calculs en lignes et en colonnes, les "copier-coller" de feuilles dès que l'on veut changer la présentation des tableaux, au risque de perdre des informations...)

Les tables.

C'est dans une table que vous allez enregistrer vos informations. Par information, on entend ici vos données.

Vous commencez par définir vos **champs**. C'est l'étape la plus importante.

- le **nom** du champ correspond à l'en-tête de colonne de votre tableau ;
- il est associé à un **type de données**, qui définit le format des données saisies dans cette colonne.

Comme type de données, vous retrouvez en particulier :

<i>Type de données</i>	<i>Définition</i>	<i>Exemples d'utilisation</i>
Texte	chaîne alphanumérique limitée à 255 caractères	Nom, Adresse, Profession, Titre, Téléphone, ...
Mémo	bloc de texte pouvant contenir jusqu'à 65536 caractères	Résumés, Informations, CV, ...
Numérique	caractères numériques, entiers ou non (le séparateur - point ou virgule - est celui défini dans les paramètres régionaux de votre panneau de configuration)	Age, Nombre d'enfants, ...
Date/Heure	données au format date (03/07/2000 par exemple)	Date d'inscription, de naissance, heure d'envoi d'un message, ...
NuméroAuto	nombre qui s'incrémente automatiquement à chaque nouvel enregistrement dans la table	Référence client, Classement, ...

Tableau VI.1. Les types de données dans une base de données.

Ensuite, vous pouvez **saisir vos données**, directement dans la table. Là vient la différence de « fonctionnement » entre tableur et base de données.

- à chaque cellule que vous remplissez, vous saisissez une **donnée** ;

- à chaque ligne que vous entrez, correspond un **enregistrement** : c'est une fiche qui contient des renseignements (données pour chaque champ) sur un élément unique de la liste ;

- chaque ligne doit être unique, et différenciable des autres : c'est à cela que sert la **clef primaire**. La clef primaire d'une table correspond au champ pour lequel toutes les données sont différentes d'un enregistrement à l'autre.

Exemple :

- si vous avez un champ "Nom de famille", et que tous les noms sont différents, alors ce champ peut servir de clef primaire ;
- par contre si vous avez des homonymes dans la liste, vous pouvez retrouver deux enregistrements pour deux personnes différentes, mais avec une même valeur "nom de famille" - là il faut trouver une autre clef primaire ;
- c'est à cela que sert le NuméroAuto : en utilisant ce type de données, on est sûr de travailler avec au moins un champ sans doublon.

- tous vos champs et vos enregistrements constituent une **table** ;

- toutes les tables composent une seule **base de données**.

Enfin, vos tables ont souvent des points communs entre-elles. Si c'est le cas, pour éviter les redondances vous allez **créer des relations**.

Les relations entre les tables.

- vous avez une table "**clients**", qui contient la référence du client, son nom, et son adresse ;
- une deuxième table, "**commande**", qui contient la référence de la commande, celle du client, la date de la commande.

Pour savoir à quelle adresse envoyer la facture de la commande, vous avez besoin de préciser que la référence du client dans une table correspond à sa référence dans l'autre table.

Démarche :

1. je prends la référence du client dans la table "**commandes**" pour la référence de la commande qui m'intéresse ;
2. je vais chercher dans la table "**clients**" l'enregistrement qui correspond à cette référence ;
3. je récupère le Nom et l'adresse du client qui correspond.

Les Requêtes.

Quand on travaille sur les bases de données, on l'a vu, les données sont enregistrées, brutes, dans des tables. **Pour y avoir accès**, pour en sélectionner quelques unes ou effectuer des calculs, on doit passer par des requêtes.

- une requête **enregistre vos critères** de sélection, ou les calculs que vous souhaitez réaliser : ainsi, à chaque fois que vous exécutez la requête, elle prend en compte les dernières informations saisies dans la table, sans y modifier aucune valeur.

- **exécuter une requête**, c'est visualiser toute ou une partie de votre table, à un instant **t**, sous certaines conditions que vous définissez vous-mêmes.

Si les requêtes sont typiques des bases de données, elles sont aussi directement **programmables en VB**.

Vous pourrez, par exemple, demander l'affichage de tous les clients habitant Oran ou les effacer de votre table à partir d'un programme en Visual-Basic (dans un module).

Les Formulaires.

Un formulaire permet de **faciliter la saisie** des données.

- un formulaire, c'est une **interface** plus ou moins soignée entre un utilisateur et la base : il permet de remplir les tables sans être directement sur la feuille de données ;

- réaliser un formulaire, c'est aussi **améliorer la saisie** des données (bulles d'aide, cellules mieux paramétrées pour recevoir les informations, ...).

Les Etats, Macros, Modules.

Sous MS-Access, vous retrouvez ces trois autres formulations : les Etats, Macros, et Modules.

- **Un état** : c'est l'équivalent d'un aperçu avant impression. Vous y définissez ce que vous voulez imprimer, et comment vous voulez l'imprimer, le logiciel vous aide en vous permettant d'**intervenir** sur la présentation, la sélection des informations. Vous pouvez même effectuer des calculs directement sur votre page.
- **Une macro** : facilite un travail répétitif.
 - une macro est une fonction qui vous permet d'**enregistrer une série d'opérations** ;
 - à chaque **exécution** de la macro, ces mêmes commandes s'exécutent, vous permettant gain de temps et d'énergie. Vous pouvez même redéfinir votre environnement de travail (barres d'outils, ...) à partir de macros.

- **Un module** : c'est un programme en Visual-Basic. Vous **programmez** vous-même vos commandes en **VB** sur un éditeur qui peut intégrer quelques assistants.

VI.9. Applications

VI.9.1. La gestion des communes.

Une application peut être réalisée afin de gérer les informations sur les communes d'Algérie.

La structure de la table qui contiendra ces informations est la suivante :

Toutes les tables		Nom du champ	Type de données
Communes_DZ		code_commune	Numérique
Communes_DZ : Table		Commune	Texte
		Wilaya	Texte
		Nature	Texte
		Code wilaya	Numérique
		Surface	Numérique
		Population_2008	Numérique

Figure VI.3. Structure de la table commune

Le contenu de la table qui comporte 1618 enregistrements, pourra être interrogé par des requêtes.

code_comm	Commune	Wilaya	Nature	Code wilaya	Surface	Population_
106	TIT	ADRAR	COMMUNE	1	1647,4063936	4417
109	TIMIMOUN	ADRAR	COMMUNE	1	9777,7139105	33060
122	BOUDA	ADRAR	COMMUNE	1	4262,0292526	9938
127	OULED AISSA	ADRAR	COMMUNE	1	4169,6981628	7685
124	TALMINE	ADRAR	COMMUNE	1	2948,1968137	12768
121	OULED AHMED	ADRAR	COMMUNE	1	4810,1546324	13547
110	OULED SAID	ADRAR	COMMUNE	1	657,39897465	8219
126	SEBAA	ADRAR	COMMUNE	1	4972,2774153	2312
117	DELDOUL	ADRAR	COMMUNE	1	1206,4637903	8647
117	DELDOUL	ADRAR	COMMUNE	1	1206,4637903	11230

Enr : 1 sur 1618 | Aucun filtre | Rechercher

Figure VI.4. Contenu de la table commune

La gestion et la mise à jour des données peuvent se faire par le biais d'un formulaire.

Communes_DZ

code_commune: 106

Commune: TIT

Wilaya: ADRAR

Nature: COMMUNE

Code wilaya: 1

Surface: 1647,40639358167

Population_2008: 4417

Figure VI.5. Formulaire pour la gestion des données de la table commune

Ci-dessous, un exemple de requête qui donne les communes de la wilaya d'Oran dont les populations recensées en 2008 sont supérieures à 100.000 habitants

Communes_DZ

- *
 - code_commune
 - Commune
 - Wilaya
 - Nature
 - Code wilaya
 - Surface
 - Population_2008

Champ :	Wilaya	Surface	Population_2008	Code wilaya	Commune
Table :	Communes_DZ	Communes_DZ	Communes_DZ	Communes_DZ	Communes_DZ
Tri :					
Afficher :	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Critères :			>100000	31	
Ou :					

Figure VI.6. Requête sur la table commune

La requête suivante effectue le calcul de la **densité de la population** des communes de la Wilaya d'Oran dont les populations recensées en 2008 sont supérieures à 100.000 habitants

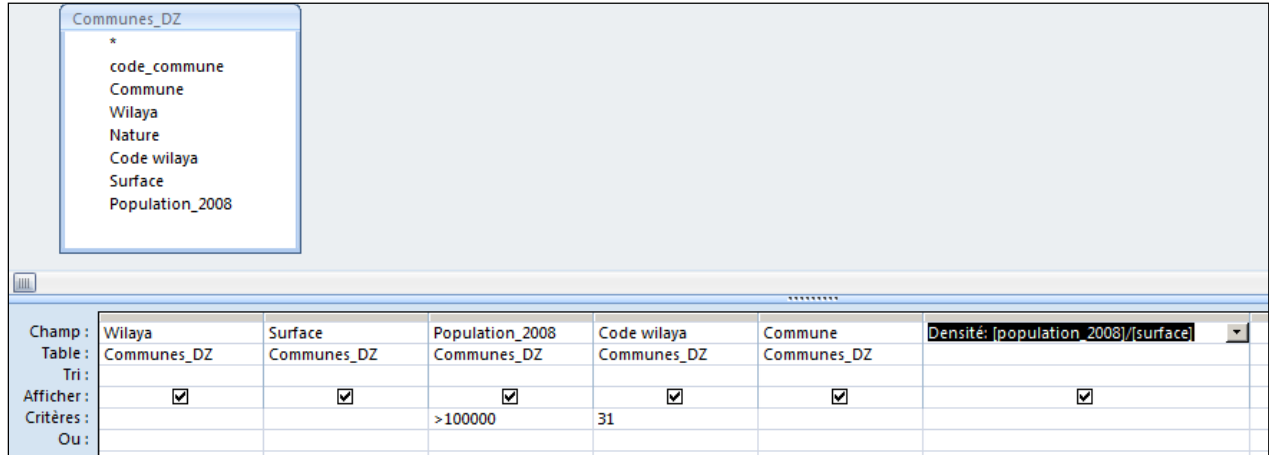


Figure VI.7. Autre requête sur la table commune

VI.9.2. La gestion des espaces.

La gestion des espaces d'un projet peut se faire à travers des grilles dont les informations sont gérées à l'aide d'une base de données.

En effet, la conception d'une construction peut reposer sur l'utilisation de grilles comme dispositifs de régularité et de neutralité telle que privilégiée par Ludwig Mies Van Der Rohe (1986).

La zone d'étude du projet est alors recouverte d'une grille gérée par une base de données comme le montre la figure VI.8.

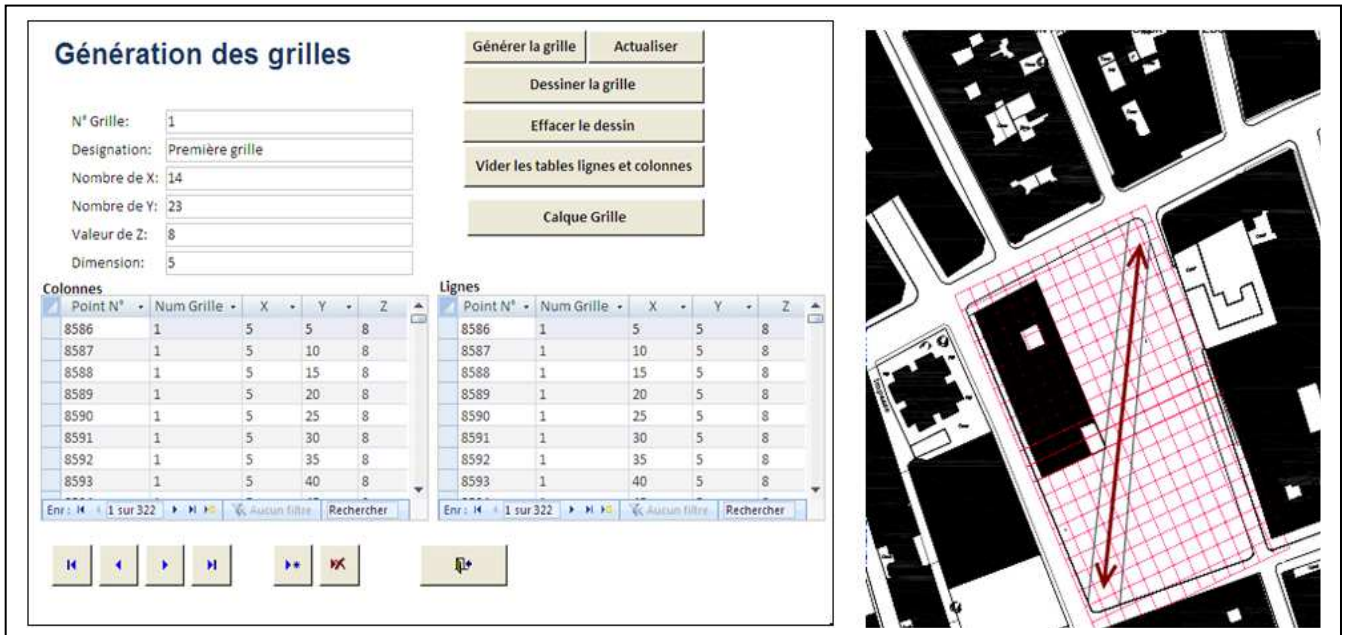


Figure VI.8. Formulaire de la base de données pour la gestion des grilles et leurs correspondances dans la zone d'étude du projet.

Source : Rahal et al., 2017.

Les espaces, constitués de mailles, sont catégorisés et supportés par des grilles régulières pour chaque niveau comme le présente la figure VI.9.

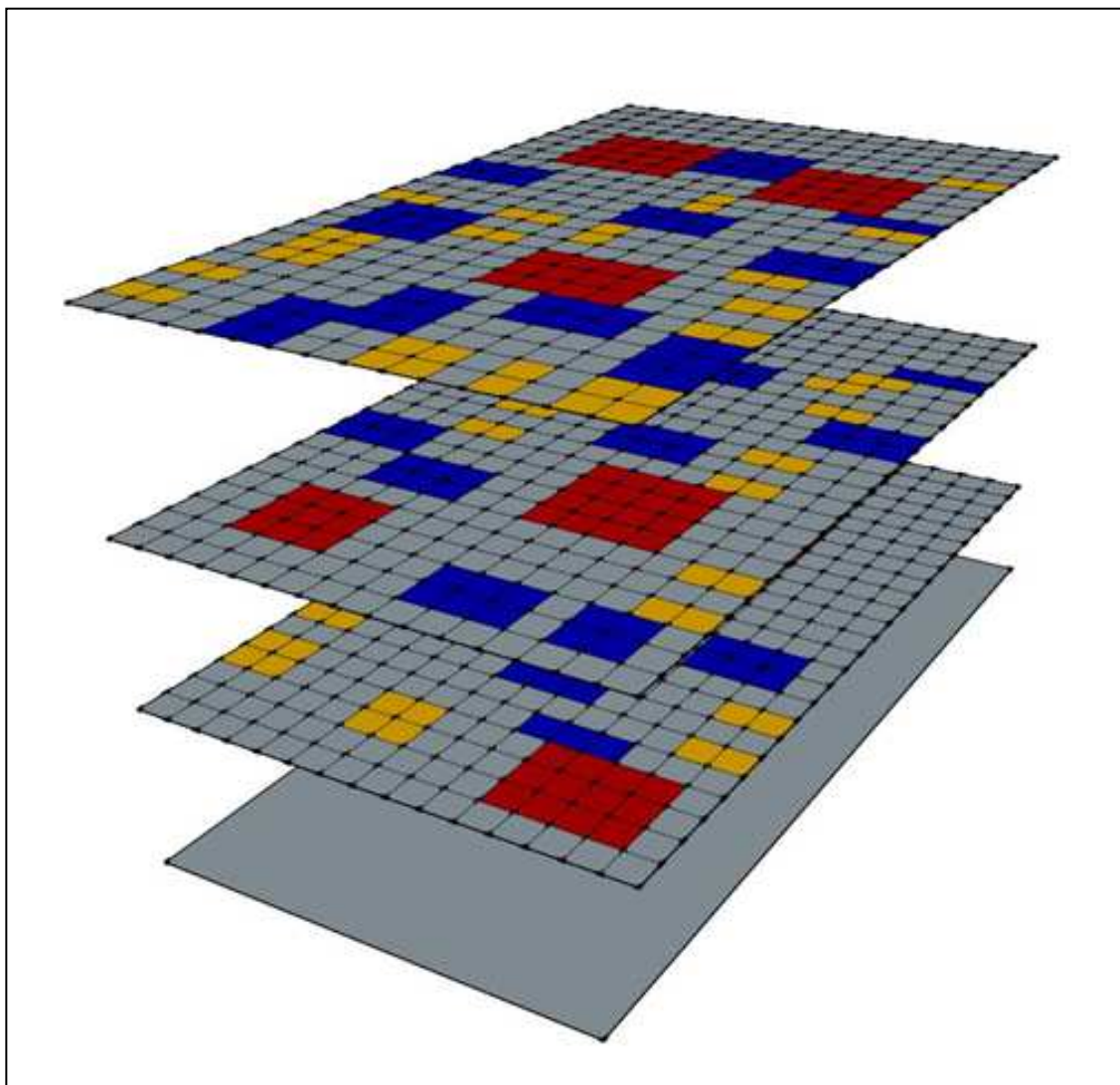


Figure VI.9. Catégorisation des espaces supportés par des grilles régulières pour chaque niveau.
Source : Rahal et al., 2017.

Pour arriver à gérer et à mettre à jour les informations de chaque grille et surtout de chaque maille, les données sont réparties sur plusieurs tables de la base de données conçue. La figure VI.10 montre la structure des tables ainsi que leurs liaisons logiques.

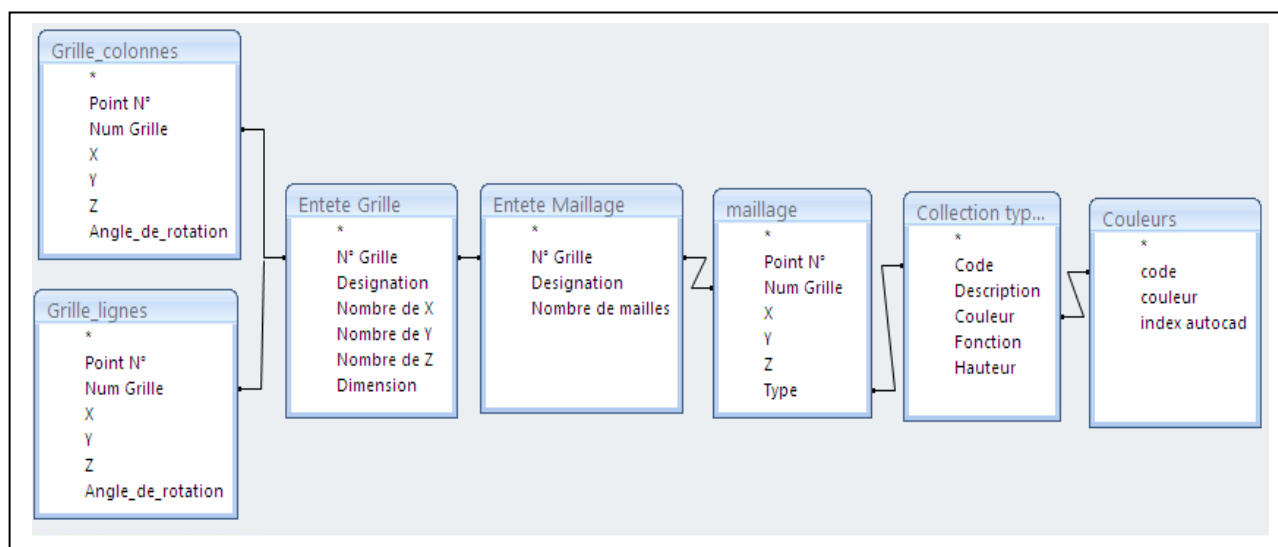


Figure VI.10. Structure et liaisons des tables de la base de données pour la gestion des espaces du projet.

Source : Rahal et al., 2017.

VI.10. Conclusion.

Actuellement, les informations deviennent de plus en plus envahissantes. Le concepteur croule sous une quantité phénoménale de données qu'il s'agit de traiter, d'analyser et d'exploiter intelligemment.

En outre, la mise à jour d'un grand nombre de paramètres dans la conception architecturale n'est pas chose aisée. Surtout si ces mises à jour sont fréquentes.

L'utilisation des bases de données, dont la capacité à transformer une grande quantité d'informations notamment par le biais de requêtes écrites en langage SQL (Structured Query Language) n'est plus à démontrer (Rahal et al., 2017).

La liaison d'une base de données avec un outil de DAO dans le cadre d'une application intégrée, permet une grande flexibilité pour la paramétrisation massive des dessins en Architecture, avec la possibilité de procéder à autant de tentatives que nécessaire afin d'aboutir à la solution qui correspond le mieux aux attentes du concepteur.

Le dessin peut être contrôlé à distance, ce qui signifie qu'il n'est pas nécessaire de faire des changements directement à partir de l'interface graphique du logiciel de DAO. En fait, les changements peuvent se faire directement à partir de la base de données (Wang et Al., 2008).

Ainsi, les bases de données peuvent constituer un support au dessin paramétrique. Cette synergie entre ces deux outils sera présentée dans le chapitre suivant.

Chapitre VII

Les bases de données et le dessin paramétrique

VII. Introduction.

L'intérêt du dessin paramétrique dans le domaine de l'architecture réside dans le fait qu'un nombre très important d'éléments de construction peuvent être regroupés dans des familles qui tendent spontanément à être paramétrées (Monedero, 2000).

Des contraintes peuvent alors être imposées sur la conception paramétrique par le concepteur lui-même pour étudier une gamme spécifique de cas. D'autres contraintes liées à l'environnement peuvent être appliquées (Taleb et Al., 2015).

L'approche paramétrique de la conception architecturale permet de répondre à ces besoins en aidant les architectes dans la génération des formes et des volumes de nature complexes qui prennent en considération des données sur différents domaines influant le projet architectural. Cependant, certains projets nécessitent la maîtrise d'un très grand nombre de paramètres mais surtout, la possibilité d'en modifier les valeurs, rapidement et facilement, selon les besoins et les contraintes qui peuvent survenir. L'utilisation des bases de données en combinaison avec les outils de CAO/DAO offre des solutions aux besoins de paramétrisation massives d'un projet.

VII.2. Liaison de la base de données et de l'outil de DAO.

Afin d'exploiter les informations contenu dans la base de données et de les traduire en dessin, il est nécessaire de lier ces 2 outils en ajoutant les bibliothèques de l'outil de dessin à l'éditeur des modules de la base de données. La figure VII.1 en présente un exemple avec Autodesk-Autocad et MS-Access.

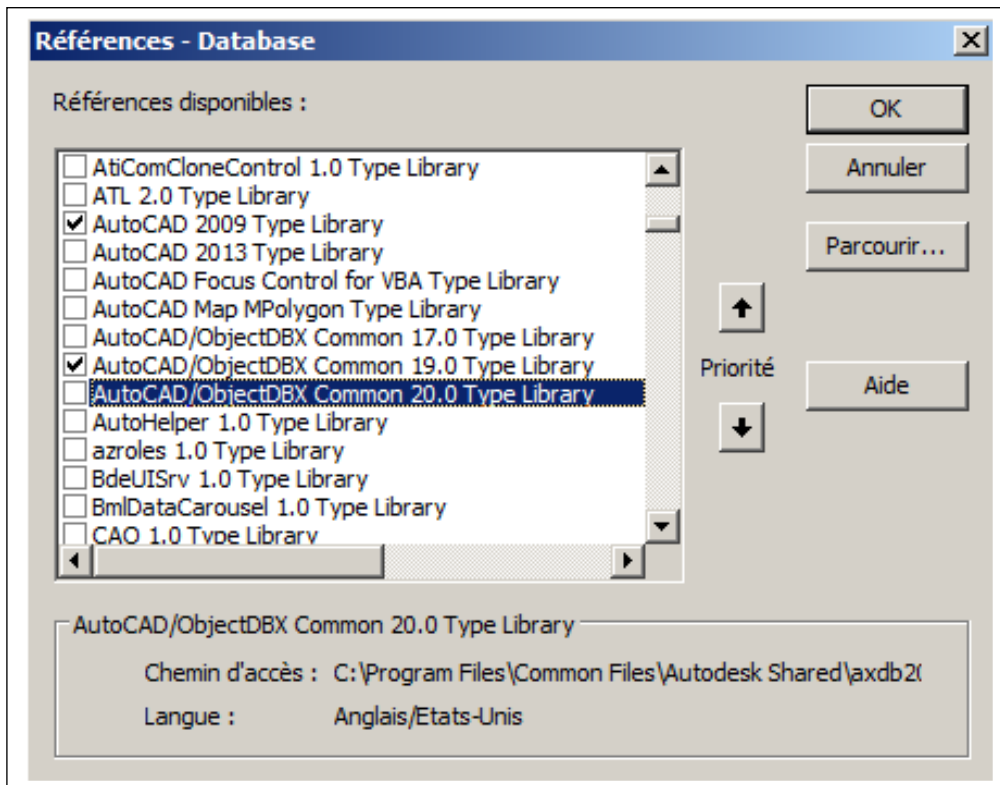


Figure VII.1. Liaison des bibliothèques d'Autodesk-Autocad sous MS-Access.

La liaison ainsi réalisée, il est possible de manipuler par programmation à la fois les données qui sont dans la base de données et les objets du dessin. La figure VII.2 montre une structure de table pour recueillir les informations sur les bâtiments, le formulaire permettant de la gérer ainsi que la traduction des données sous forme de dessins.

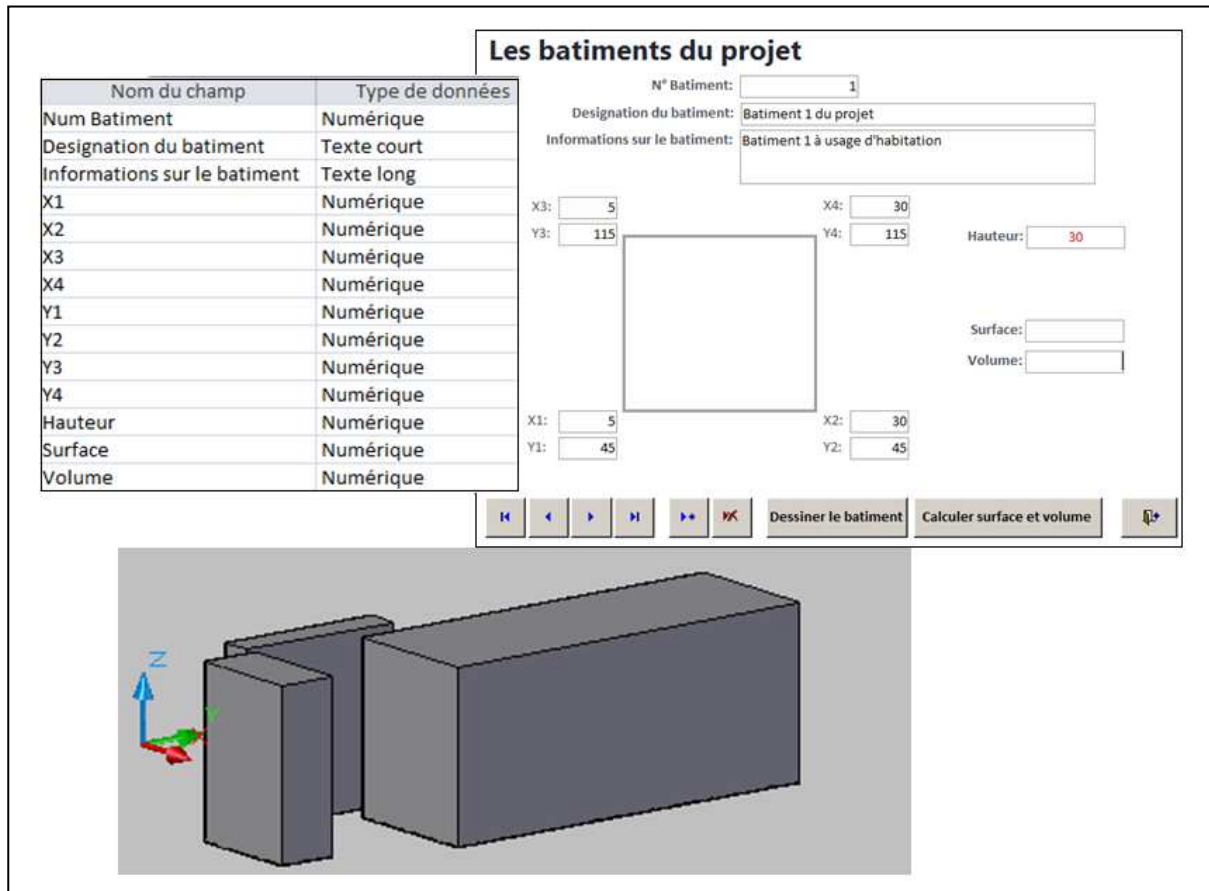


Figure VII.2. Structure de table, formulaire dédié et résultat sous forme de dessin.

VII.3. Partage des données.

La mise en partage de la base de données soit sous un réseau local ou un réseau longue distance, permet à plusieurs intervenants de participer à un même projet. Selon les privilèges alloués par le chef de projet, les participants pour accéder aux informations de projet en mode lecture seule, modification ou ajout.

La figure VII.3 montre des bâtiments réalisés par participants à travers un réseau local.

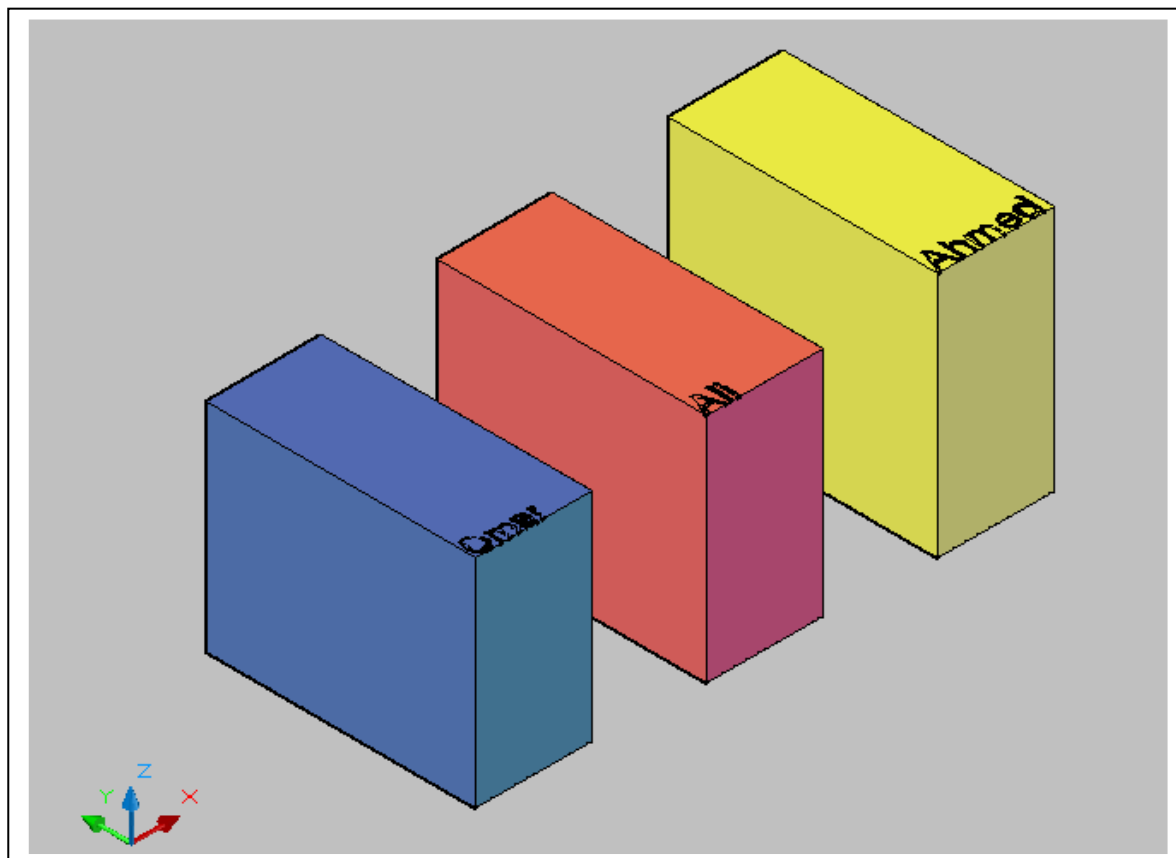
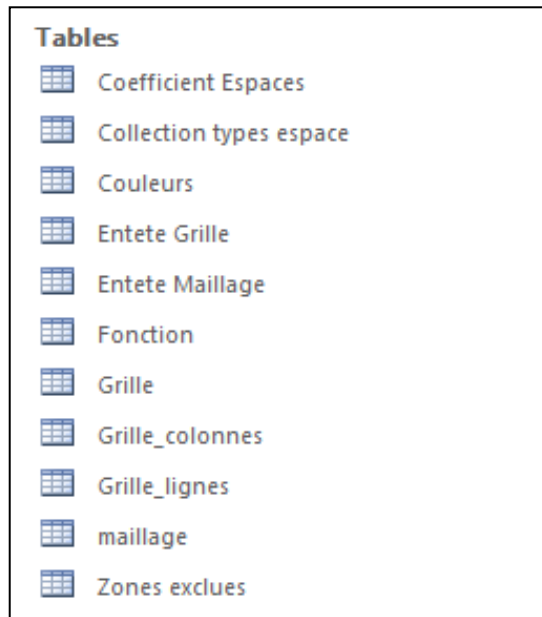


Figure VII.3. Objets réalisés par plusieurs intervenants à travers un réseau local.

VII.4. Application.

L'application de cette méthodologie à la gestion des espaces et à leurs connexions se fait à la fois à l'aide de tables dédiées à chaque fonctionnalité et à l'aide de programme exploitant les informations de la base de données et les mettant à jour selon les besoins exprimées par le concepteur.

La figure VII.4 montre les tables de la base de données pour collecter les coordonnées tridimensionnelles des éléments constitutifs de chaque grille dont les données sont générées automatiquement en fonction des informations de base.



The image shows a screenshot of a database interface with a list of tables. The title 'Tables' is at the top left. Below it, there are 12 entries, each consisting of a small grid icon followed by the table name. The table names are: Coefficient Espaces, Collection types espace, Couleurs, Entete Grille, Entete Maillage, Fonction, Grille, Grille_colonnes, Grille_lignes, maillage, and Zones exclues.












Tables	
	Coefficient Espaces
	Collection types espace
	Couleurs
	Entete Grille
	Entete Maillage
	Fonction
	Grille
	Grille_colonnes
	Grille_lignes
	maillage
	Zones exclues

Figure VII.4. Tables de la base de données dédiée à la gestion des espaces.

Que ce soit dans un bâtiment ou en milieu urbain, les grilles conçues seront un support à l'affectation des espaces à un type d'usage.

La génération de ces espaces, qui sont constitués d'un ensemble de mailles, et leurs affectations pourront se faire aléatoirement afin d'aider l'architecte à explorer toutes les possibilités offertes par le hasard qui sont contraintes et façonnées par des limitations.

Par conséquent, la production d'options alternatives d'aménagement de l'espace nécessite au départ de créer des contraintes qui sont gérées par programmation. La figure VII.5 montre le formulaire qui permet la mise à jour à la fois des données et des dessins.

Espaces

Vider la table | Dessin des zones d'exclusion | Générer les espaces en spirales

Actualiser | Dessin des espaces

Générer les espaces en rectangles

Effacer le dessin | Calque Maillage

Grille Première grille N° Grille 1 Nb X 14 Nb Y 23 Nb Z 8 Dimension 5

Designation du maillage: Tentative N°1 Nombre de mailles: 300

Les coordonnées du maillage généré

Point N°	Num Grille	X	Y	Z	Type	Num Espace
3056649	1	15	15	8	2	227
3056650	1	15	20	8	2	227
3056651	1	15	25	8	2	227
3056648	1	20	15	8	2	227
3056645	1	20	20	8	2	227
3056652	1	20	25	8	2	227
3054703	1	30	25	8	3	1
3054704	1	30	30	8	3	1
3054705	1	30	35	8	3	1
3054702	1	35	25	8	3	1

Enr : 1 sur 34 Non filtré Rechercher

⏪ ⏩ ⏴ ⏵ ⏴* ✖ ↻

Figure VII.5. Formulaire pour la gestion des données et des dessins liés aux espaces conçus.

La base de données liée aux techniques du dessin paramétrique joue un rôle clé dans la production d'un grand nombre de solutions qui dépendent elles-mêmes d'un grand nombre de paramètres. Le fait qu'une seule application gère à la fois la base de données et l'outil de dessin facilite le travail du concepteur et lui permet d'effectuer facilement autant de combinaisons que nécessaire pour choisir la solution la plus appropriée.

Le résultat apparaît dans la figure VII.6 sous forme de dessin, de plan et de volume.



Figure VII.6. Le résultat de la simulation sous forme de dessin, de plan et de volume.

Source : Rahal et al., 2018.

VII.5. Conclusion.

La liaison d'une base de données avec un outil de DAO dans le cadre d'une application intégrée, permet une grande flexibilité pour la paramétrisation massive des dessins en Architecture, avec la possibilité de procéder à autant de tentatives que nécessaire afin d'aboutir à la solution qui correspond le mieux aux attentes du concepteur (Rahal et al., 2017).

Pour l'Architecte, la capacité de modélisation d'un nombre presque infini de possibilités virtuelles, offre un potentiel intéressant de suggestions et d'imaginations de solutions spatiales et de constructions innovantes (Silvestri, 2009). L'utilisation du hasard dans la combinaison des valeurs des différents paramètres de la base de données a permis d'explorer un nombre très important de solutions en éliminant automatiquement celles qui ne correspondent pas à la vision de l'architecte.

Cette méthodologie est non seulement utile dans la phase de conception mais aussi dans les phases de transformation et de réadaptation mettant en jeu un nombre important de paramètres.

Une base de données partagée et exploitée par les techniques du dessin paramétrique correspond aux impératifs de la démarche BIM dont le prochain chapitre y sera consacré.

Chapitre VIII

La démarche BIM

VIII.1. Introduction.

Le BIM (Building Information Modeling) n'est pas un logiciel, un outil ou un programme. Le BIM est une démarche qui a pour but d'impulser une nouvelle manière de concevoir, de construire et d'appréhender la collaboration entre les parties prenantes d'un projet de construction.

Apparu au début des années 90, le BIM avait pour objectif de définir la représentation numérique des constructions sous la forme d'un modèle 3D enrichi de propriétés. Ce sont ces mêmes propriétés qui constituent les données et que l'on retrouve dans tous les objets de la maquette numérique.

La maquette numérique est constituée d'objets tels que les murs, les portes ou les fenêtres. Ces objets contiennent des propriétés telles que la hauteur, la largeur, le poids ou le matériau. Alors, qu'un dessin en 2D ou en 3D est dépourvu de ce type d'information.

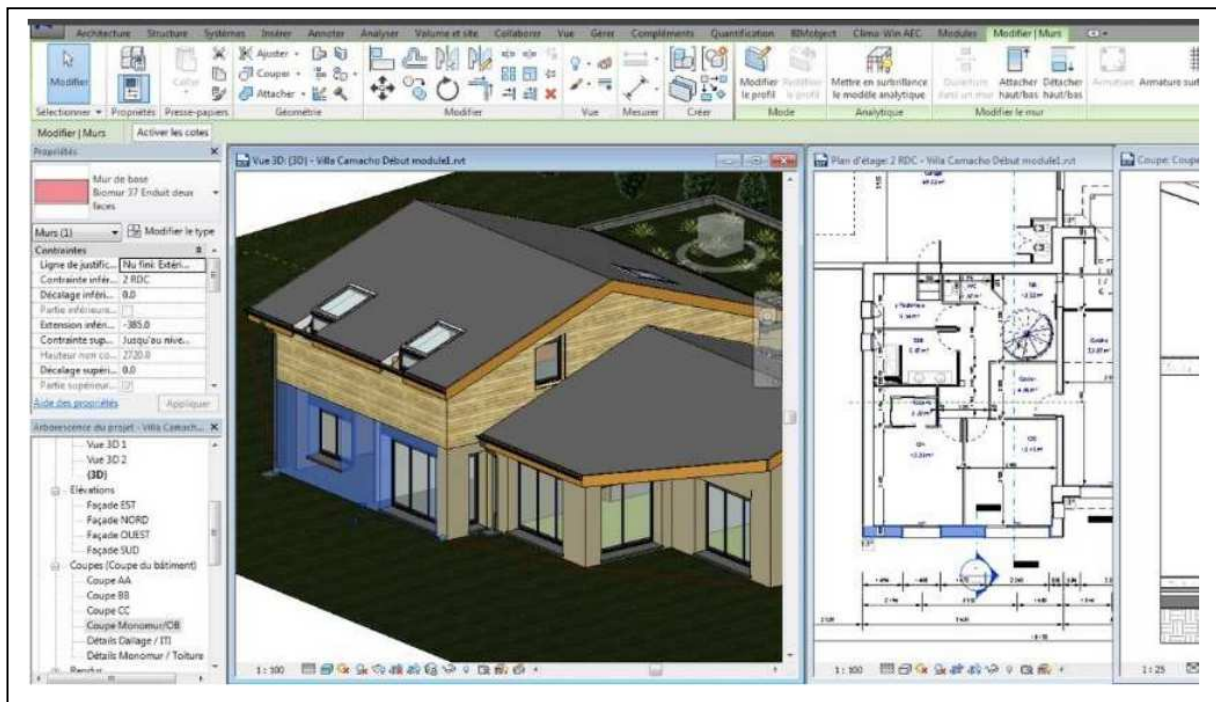


Figure VIII.1. Exemple d'un projet réalisé avec un outil BIM propriétaire

Source: Autodesk

Par conséquent, la donnée est une notion fondamentale dans la démarche BIM dont le cœur est une base de données, soit propriétaire, telle que celle de Revit ou bien personnelle si le concepteur développe sa propre démarche.

VIII.2. Les objectifs du BIM.

Le principal objectif de la démarche BIM est d'accompagner le projet depuis la phase de conception jusqu'à sa démolition. En d'autres termes, gérer le projet de construction durant toute sa durée de vie.

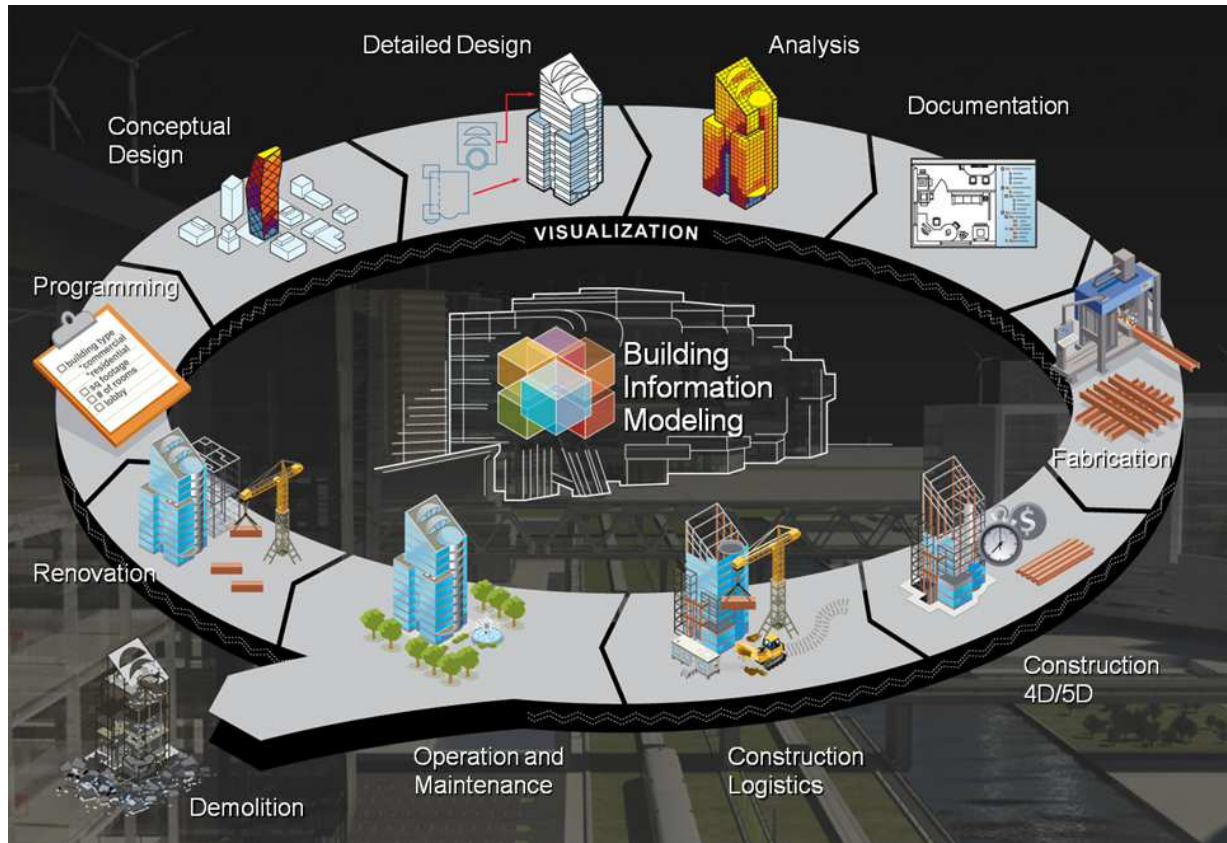


Figure VIII.2. Le cycle de vie de la démarche BIM

Source: Autodesk

Avec l'idée de regrouper et de partager les informations, le BIM permet de maintenir une continuité à travers toute la vie de l'ouvrage.

VIII.3. Les avantages du BIM.

Le BIM intervient sur l'ensemble du cycle de vie du bâtiment. À chaque phase, les maquettes sont différentes, tant par leurs qualités graphiques que par les données qu'elles intègrent. L'intérêt du BIM est de pouvoir anticiper les erreurs, détecter les conflits et diminuer les risques que ce soit à travers l'analyse des collisions entre deux maquettes ou bien avec une meilleure coordination des équipes du projet.

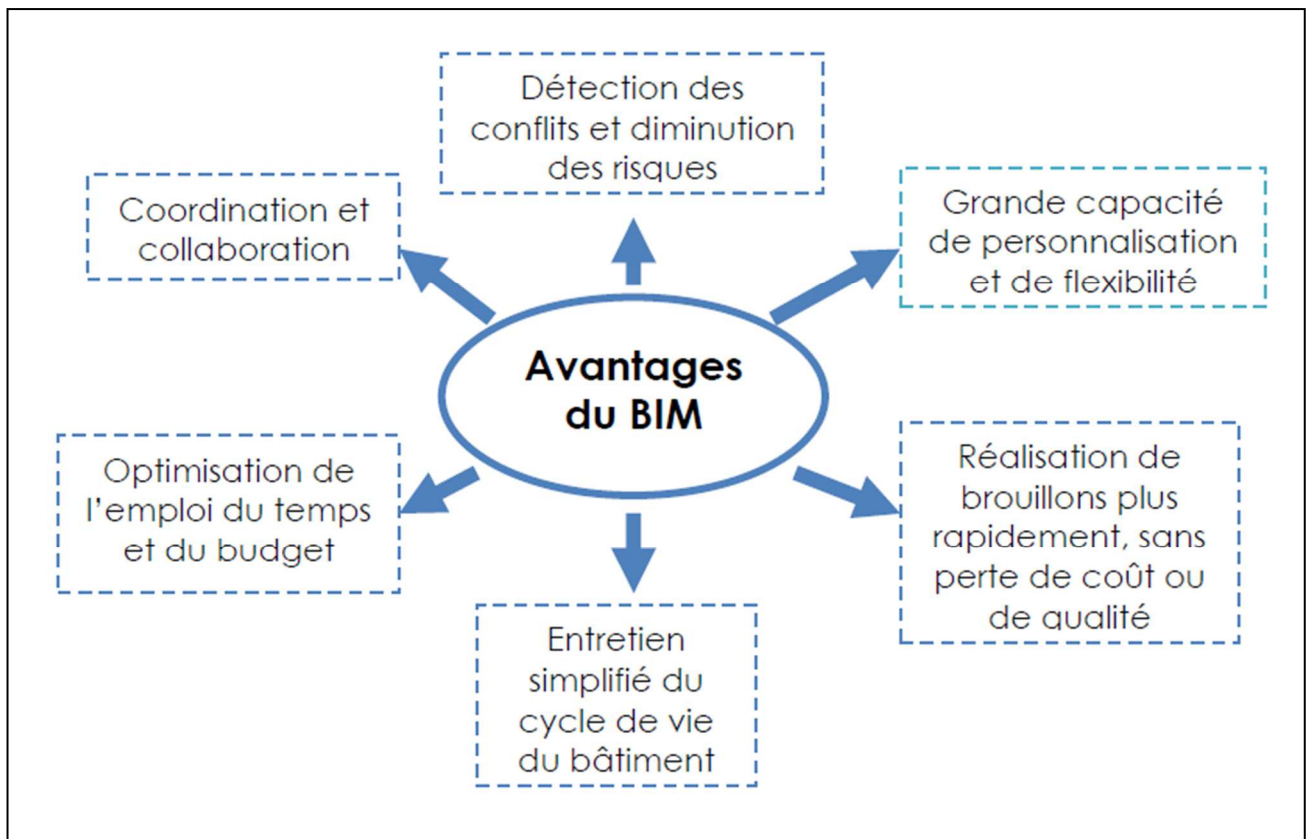


Figure VIII.3. Les avantages du BIM.

Il est évident que les différents intervenants du projet ne sont pas intéressés par le BIM de la même manière, mais chacun y trouve son intérêt.

En phase de conception, le maître d'œuvre peut mieux s'approprier le projet grâce aux visuels générés par la maquette. La communication avec des non-professionnels tels que les citoyens ou les usagers est ainsi facilitée et peut être un support à l'urbanisme participatif.

Il est alors possible d'intégrer et de simuler plus rapidement et plus facilement différents scénarios afin de répondre aux exigences du programme et aux contraintes techniques.

De plus, dès la phase de conception, le maître d'œuvre prend en compte et intègre les besoins en exploitation du maître d'ouvrage dans les maquettes. Cela se passe notamment au moment de la consultation des entreprises qui peuvent anticiper les besoins en maquette numérique qu'elles auront en phase d'exécution.

En phase de construction, les processus administratifs et financiers sont optimisés grâce à la structuration de l'information et à l'attribution des tâches.

Les erreurs sur chantier sont également anticipées, car les entreprises disposent de maquettes sans conflits qui ont été préalablement détectées et corrigées. Cela contribue et faire diminuer les réserves.

VIII.4. Les outils propriétaires du BIM.

Il existe différents types d'outils de modélisation par corps d'état, qui n'auront pas les mêmes objets ni les mêmes paramétrages d'objets. Les principales catégories que l'on rencontre dans les outils de modélisation concernent l'architecture, les structures, l'électricité, le chauffage, ventilation et climatisation (CVC).

Ces outils de modélisation peuvent être complétés par des plug-ins pour faciliter la modélisation paramétrique et l'analyse du modèle.

Ces outils sont en général utilisés au début de la chaîne de conception et prennent en charge toutes les phases du projet et impliquent tous les intervenants.

Par exemple, ces outils permettent dans la partie structure le calcul de la section des poteaux et des poutres, les dimensions des fondations. Dans la partie électricité, le calcul de la section des câbles, le calcul du facteur de lumière du jour et des nombre et type de luminaires qui en découlent. Dans la partie thermique, le dimensionnement des gaines, des groupes de chauffage ou de climatisation.



Figure VIII.4. Quelques outils BIM propriétaires.

La performance des outils BIM propriétaires est indiscutable, cependant, dans les pays en développement, une barrière économique est érigée par le coût de leurs licences d'utilisation limitées dans le temps.

Figure VIII.5.. Le système d'abonnement d'un outil BIM propriétaire

VIII.5. Le format IFC.

Pour faciliter la coopération entre les différents acteurs du bâtiment et rationaliser les méthodes de travail le format IFC (Industry Foundation Classes) a été créé par l'International Alliance for Interoperability (I.A.I) devenue plus tard building Smart International (B.S.I).

Ainsi, les logiciels BIM peuvent échanger dans un format commun. Les visualiseurs BIM permettent d'ouvrir des fichiers IFC échangés sur un projet et de récupérer des informations.

Ces outils peuvent, eux aussi, être utilisés par différents contributeurs pour différents usages.

Ils permettent aux intervenants du projet de pouvoir lire des informations issues des maquettes numériques sans avoir à investir dans des logiciels, des ordinateurs et des formations.

C'est un format qui est aussi utilisé par les outils libres du BIM.

VIII.6. Les outils libres du BIM.

Une approche du BIM libre, ouvert et universel, permet aux différents acteurs d'un projet de partager leurs données en utilisant des outils BIM gratuits.

Certes, les performances de ce type d'outils sont moindres que les outils BIM propriétaires des grands éditeurs de logiciel, mais cela reste une solution intéressante permettant l'accès à cette technologie devenue incontournable dans les projets de construction.



Figure VIII.6. Quelques outils BIM libres

VIII.7. Une autre alternative aux outils BIM propriétaires.

Une autre alternative à la contrainte de la barrière économique réside dans la création d'un système d'information propre à un projet ou un ensemble de projets prenant en considération leurs spécificités tout au long de la durée de vie du bâtiment.

Cette approche permet de prendre en charge ces spécificités avec la possibilité de développer une application permettant le pilotage, à la fois de la base de données et de l'outil de CAO/DAO.

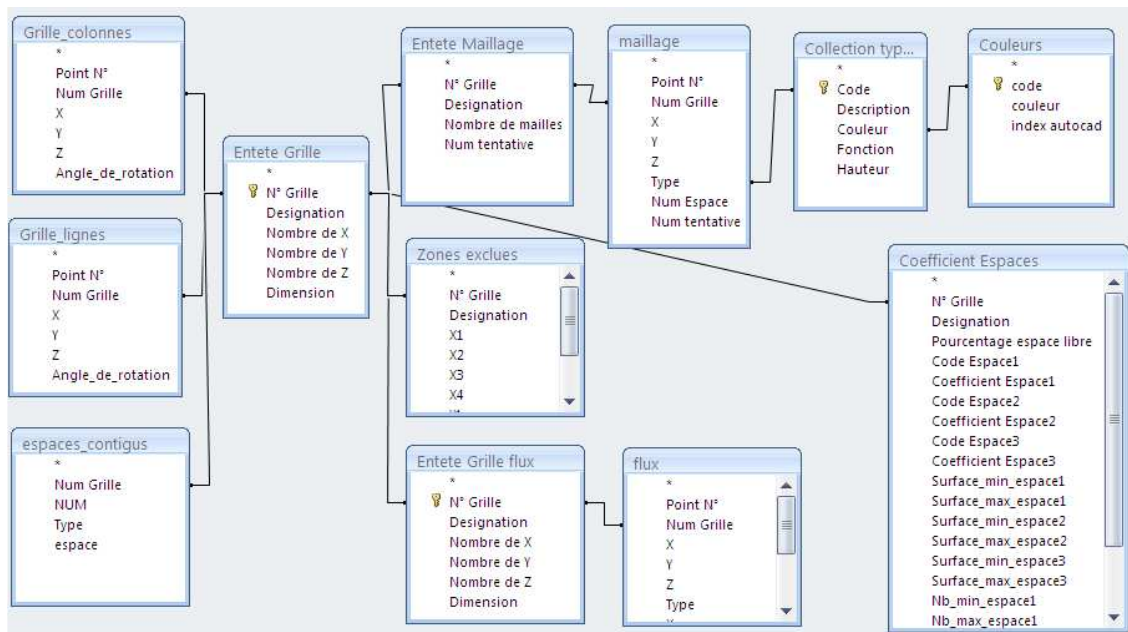


Figure VIII.7. L'architecture du système d'information d'un projet lié à un outil de CAO/DAO

Source: Rahal et al., 2018.

Le système d'information peut être mis en réseau et des privilèges de lecture, modification ou ajout sont affectés aux acteurs du projet selon le domaine d'intervention de chacun. L'application développée deviendrait alors collaborative, évolutive et efficace, permettant la transformation instantanée des données traitées en dessins ou en objets du projet.

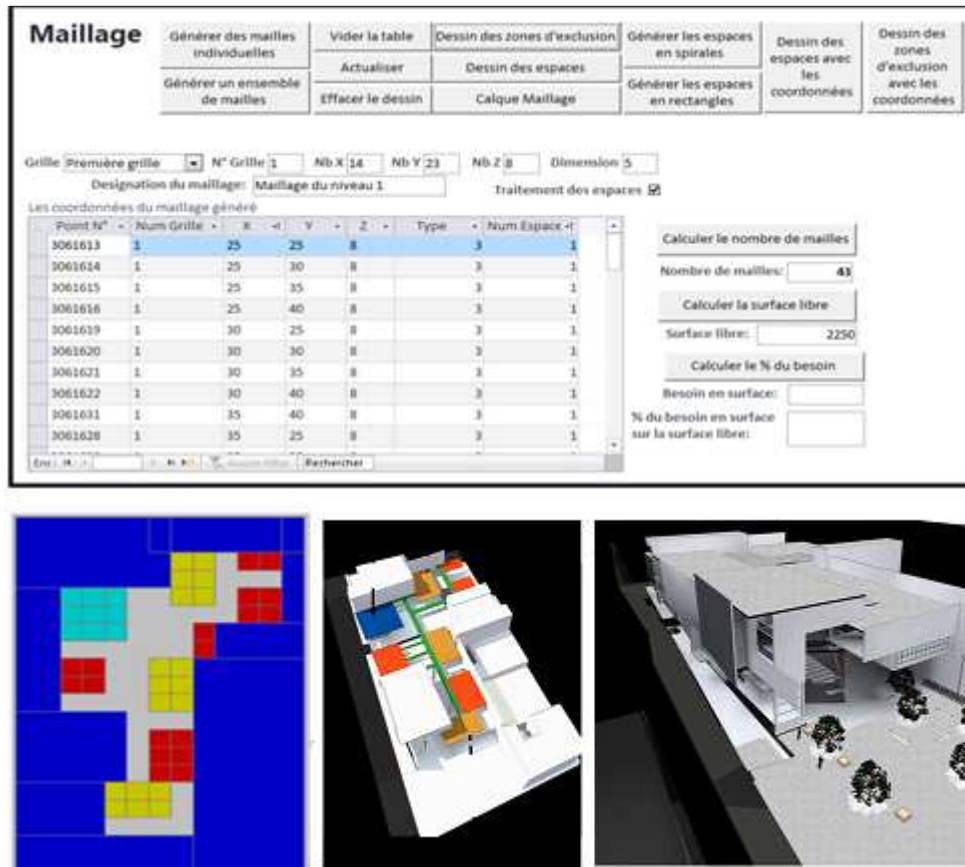


Figure VIII.8. Exemple d'un projet conçu à l'aide d'une base de données lié à un outil de CAO/DAO

Source: Rahal et al., 2017.

VIII.8. Conclusion.

Les outils de la technologie BIM (Building Information Modeling) rencontrent des succès dans cette approche numérique.

Cette approche permet la formation de compositions complexes, à la fois formelles et conceptuelles, à travers la mise en œuvre d'un ensemble simple d'opérations et de paramètres (Dino, 2012). L'exemple est donné par les derniers outils BIM (Building Information Modeling) qui incluent Autodesk Revit, Gehry's Digital Technology Project, Geometric Component, Rhinoceros, Grasshopper et ParaCloud (Chen et al., 2015).

Le BIM est une méthodologie de travail collaboratif pour la création et la gestion d'un projet de construction. L'objectif est de centraliser toutes les informations du projet dans une maquette numérique, qui conforme une grande base de données qui permet la gestion de tous les éléments de l'infrastructure tout au long de son cycle de vie. Il s'agit d'une évolution par rapport aux systèmes de conception traditionnels basés sur des plans, intégrant des informations géométriques, des temps, des coûts, l'environnement et la maintenance. L'un des principaux objectifs de la méthodologie BIM est de travailler efficacement, en essayant d'optimiser toutes les activités qui composent un projet, puis d'en réduire la durée et d'augmenter la productivité (Troncoso-Pastoriza et al., 2018).

Cependant, certains projets nécessitent la maîtrise d'un très grand nombre de paramètres mais surtout la possibilité de changer les valeurs rapidement et facilement, en fonction des besoins et de l'apparition de contraintes.

La solution serait une base de données dédiée à la gestion de toutes les informations du projet, et qui serait automatiquement lié à un outil de dessin assisté par ordinateur par le biais des techniques du dessin paramétrique. Le système d'aide ainsi conçu peut donner souplesse et adaptabilité aux projets impliquant un grand volume d'informations.

Conclusion générale.

Répondre à l'énorme demande de structures construites en raison de l'augmentation de la population et de la croissance des économies n'est pas une tâche facile et ne peut certainement pas être fait avec l'approche conventionnelle (Sharma et al., 2017). Au fur et à mesure que notre connaissance du mécanisme de production architectural change, la conception de l'architecture ne cesse de changer aussi (Zhu et al., 2015).

L'architecture paramétrique, en tant que manière de penser les bâtiments comme un ensemble d'aspects codés numériquement, démontre les possibilités d'utiliser des ressources mathématiques pour améliorer le fonctionnement du bâtiment (Czech et al., 2016).

Les bâtiments ne peuvent être réduits à leur seule géométrie, ils incluent également des connaissances. De nouvelles pistes de recherche émergent pour passer d'un raisonnement purement géométrique à un niveau sémantique de représentation numérique des formes (De Luca et al., 2005).

En conséquence, et compte tenu de la complexité des décisions et de la diversité des données à gérer, nous proposons une solution basée sur un système d'information qui aide l'architecte dans l'allocation des espaces de son projet. La contribution du système développé au projet est avant tout l'aide apportée à une conception d'espaces conforme à la vision de l'architecte, même si elle est complexe. Cette solution pourrait être utilisée dans d'autres domaines de l'architecture où le concept de grille est nécessaire comme support. Cette étude était limitée à un projet architectural cependant, l'utilisation de la grille permet de changer d'échelle et de s'intéresser aux projets urbains.

Le système combinant les bases de données et les techniques du dessin paramétrique joue un rôle clé dans la production d'un grand nombre de solutions qui dépendent elles-mêmes d'un grand nombre de paramètres gérés par le système d'information. Le fait qu'une seule application gère à la fois le système d'information et l'outil de dessin facilite le travail du concepteur et lui permet d'effectuer facilement autant de combinaisons que nécessaire pour choisir la solution la plus appropriée.

En outre, le système d'information peut évoluer pour prendre en compte d'autres aspects qualitatifs des espaces ou des bâtiments tels que la lumière, la ventilation ou la perception.

Enfin, la mise en réseau du système d'information peut permettre à plusieurs acteurs de contribuer dans le cadre d'un grand projet. A l'instar des outils BIM, cette méthodologie peut fournir une plate-forme robuste pour la communication et le partage d'informations entre toutes les parties prenantes d'un projet (Akinade et al., 2017), réduire les erreurs de conception et augmenter la productivité. Des privilèges peuvent être attribués à différents concepteurs par domaine d'intervention de chacun. Le système deviendrait ainsi, collaboratif, évolutif et efficace.

Références bibliographiques

Akinade O. O., Oyedele L. O., Omoteso K., Ajayi S. O., Bilal M., Owolabi H. A., Looney J. H. (2017), BIM-based deconstruction tool: Towards essential functionalities, *International Journal of Sustainable Built Environment* 6 (1): 260-271.

Bressani, M., Carpo, M., Martin, R., Picon, A., & Vardouli, T. (2019). L'architecture à l'heure du numérique, des algorithmes au projet. Un débat entre Martin Bressani, Mario Carpo, Reinhold Martin et Theodora Vardouli, mené par Antoine Picon. *Perspective. Actualité en histoire de l'art*, (2), 113-140.

Bravo, M. (2013), in M. A. Schnabel (ed.), *Cutting Edge: 47th International Conference of the Architectural Science Association*, pp. 147–156. © 2013, The Architectural Science Association (ANZAScA), Australia

Cache, B. (2013), "Dürer - Vitruvius - Plato. Instruments of Thought », <http://architectureinthemaking.se/output/lectures/cache>. Lecture held at the Event: Environment Review Days organized by architectureinthemaking. Chalmers Department of Architecture KTH School of Architecture, Architecture and Built Environment, LTH, Umeå School of Architecture

Chareb-Yssad I., (2016). *Le Dessin Assisté par Ordinateur*. Université Aboubekr Belkaid Tlemcen

Chen Z. R., Lim C. K., Shao W. Y. (2015), Comparisons of practice progress of digital design and fabrication in free-form architecture, *Journal of Industrial and Production Engineering* 32(2): 121-132.

Czech A., Borucka J. (2016), *The Use of the Language of Mathematics as an Inspiration for Contemporary Architectural Design*, *Procedia Engineering* **161**: 1582-1587.

Delvaux, F., & Marin, P. (2016). La logique paramétrique, héritage d'une science de la variation pour un renforcement de la connaissance en architecture.

De Luca L., Véron P., Florenzano M. (2005), *Modélisation sémantique et multi-représentation en architecture*, In Int. Conf. Virtual Retrospect, Biarritz, pp. 8-10

Dino I. (2012), Creative design exploration by parametric generative systems in architecture, METU Journal of Faculty of Architecture 29(1): 207-224.

Jormakka, K., Schürer, O. (2007). *La recherche de la Forme*, Birkhauser. Migayrou, F. (2003). Préface. In *Architectures non standard*. Editions du Centre Pompidou

Monedero, J. (2000). Parametric design: a review and some experiences. *Automation in Construction*, 9(4), 369-377.

Rahal, F., Hadjou, Z. (2018). Information system for parametric architecture, dedicated to spaces allocation. *Urbanism Architecture Constructions*, 9(4), 337-346.

URL : http://uac.incd.ro/Art/Art_UAC_258_final.pdf

Rahal, F., Hadjou, Z. (2017). The massive parametrization of architectural design: The case of informal space. *Algerian Journal of Engineering Architecture and Urbanism*, Vol. 1, Nr.2.

URL : www.aneau.org/ajeau/index.php/archives-of-research-papers/vol-1-issue-no-2-2017/v1n2a04.html

Sharma S., Sawhney A., Arif M. (2017), *Parametric Modelling for Designing Offsite Construction*, *Procedia Engineering* **196**: 1114-1121.

Silvestri, C. (2009). Perception et conception en architecture non standard: une approche expérimentale pour l'étude des processus de conception spatiale des formes complexes Thèse de doctorat. Montpellier 2.

Taleb, H., Musleh, M. A. (2015). Applying urban parametric design optimisation processes to a hot climate: Case study of the UAE. *Sustainable Cities and Society*, 14, 236-253.

Tourpe, A. (2004). Le Dessin Assisté par Ordinateur (DAO) dans la formation des ingénieurs. Université de Louvain. Belgique.

Troncoso-Pastoriza F., Eguía-Oller P., Díaz-Redondo R. P., Granada-Álvarez E. (2018), Generation of BIM data based on the automatic detection, identification and localization of lamps in buildings, *Sustainable Cities and Society* 36: 59-70.

Turrin, M., Von Buelow, P., Kilian, A., Stouffs, R. (2012). Performative skins for passive climatic comfort: A parametric design process. *Automation in Construction*, 22, 36-50.

Van Der Rohe, L. M. (1986). *The Mies Van Der Rohe Archive (Vol. 18)*. Taylor & Francis.

Wang, S. H., Melendez, S., Tsai, C. S. (2008). Application of parametric sketching and associability in 3D CAD. *Computer-Aided Design and Applications*, 5(6), 822-830.

Zhu L., Zhang N., Qing X. (2015), *Research on Algorithm Schema of Parametric Architecture Design Based on Schema Theory*, *Advanced in Intelligent Systems Research* 126: 576