



UNIVERSITÉ SAAD DAHLAB

MÉMOIRE DE FIN D'ÉTUDE MASTER II

**Etude et développement d'un éditeur
graphique en 3D pour un système de
contrôle, de supervision et
d'aquisition des données**

Présenté par :

- Belkhos Ikbal Wahib.
- Hammouche Anes.

Promoteur :

- Mr. Cherif Zahar
Mohammed Amine.

Encadreuse :

- Mme. Kello Ratiba.

2019-2020

Remerciements

Tout d'abord, nous remercions notre Dieu le tout puissant de nous avoir donné la force d'atteindre notre objectif et accomplir notre travail.

*Nous voudrions dans un premier temps remercier, notre promoteur **Mr. Cherif Zahar Mohammed Amine** pour ses patiences, ses disponibilités et surtout ses judicieux conseils, qui nous ont aidé à bien mener ce mémoire.*

*Nous remercions également à Madame **Abed**, qui a accepté de présider et d'honorer de sa présence le jury de soutenance du présent mémoire de Master. Qu'il trouve ici l'expression de notre reconnaissance et de notre profond respect.*

*Notre remerciements vont également aux membres du jury Madame **Chikhi I.** pour l'honneur qu'elle nous a fait en participant à l'évaluation de ce travail.*

*Nous remercions aussi l'encadreuse de l'entreprise **Mme. Kello Ratiba**, qui a partagé ses connaissances et leurs expériences dans ce milieu, tout en nous accordant leurs confiance et une large indépendance dans la préparation de ce mémoire.*

Enfin, nous remercions aussi nos parents et nos amis qu'on n'a pas cité et qui ont toujours été là pour nous. Leurs soutien inconditionnel et leurs encouragements ont été d'une grande aide.

Dédicace

Je dédie ce fruit de mes longues années d'études tout d'abord :

Ames très chers parents, qui sont la lumière de ma vie, qui ont tant souffré et sacrifiés pour que je sois heureux, pour leurs conseils, leur affection et leurs encouragements.

Je vous remercie pour tout vos efforts fournis pour moi, que Dieu vous garde, vous protège, et vous bénisse la vie.

Et je le dédie :

Atoutes les membres de ma famille, mon frère « Sohaib », mes deux sœurs « Asmaa » et « Lina ». A toutes mes proches « Bachir », « Nacer », « Abdelatif » et « Zakaria ».

Atoutes mes amis et mes collègues avec qui j'ai partagé de très bons moments tout le long de ces années.

Atoutes les enseignants et les collègues de la promotion de la deuxième année Master 2019/2020.

Hammouche

Anes

Dédicace

Je dédie ce fruit de mes longues années d'études tout d'abord :

Ames très chers parents, qui sont la lumière de ma vie, qui ont tant souffert et sacrifiés pour que je sois heureux, pour leurs conseils, leur affection et leurs encouragements.

Je vous remercie pour tout vos efforts fournis pour moi, que Dieu vous garde, vous protège, et vous bénisse la vie.

Et je le dédie :

Atoutes les membres de ma famille, mes soeurs « Raouda », « Kaoutar », « Nafisa » et « Ikram ». A toutes mes proches « Khalil », « Walid » et « Rania ».

Atoutes mes amis et mes collègues avec qui j'ai partagé de très bons moments tout le long de ces années.

Atous les enseignants et les collègues de la promotion de la deuxième année Master 2019/2020.

**BELKHOS
IKBAL WAHIB**

Résumé

Dans le cadre de notre travail, on intéresse au système de contrôle, de supervision et d'acquisition des données SCADA qui est un système de supervision qui traite un grand nombre de mesures et contrôle à distance les installations. Les éditeurs graphiques 2D sont des éditeurs pour la modélisation 2D, mais comme nous savons tous que la modélisation 2D ne représente pas l'image réelle de la région électrique. Par conséquent, l'objectif de cette étude est de développer un éditeur graphique 3D qui sert à construire des plans d'électricité en 3D.

Pour répondre à ces besoins, on a utilisé des outils de modélisation 3D (SketchUp et Blender) pour construire les composants nécessaires, et on a implémenté ces composants en JavaFX en utilisant la bibliothèque "Interactivemesh".

Dans ce mémoire, on a parlé sur les étapes nécessaires pour la réalisation de cet éditeur, les outils utilisés au cours de la réalisation de l'éditeur.

ملخص

في إطار المشروع الذي نقوم به، نحن مهتمون بنظام حفظ البيانات والإشراف عليها وكذا الحفاظ عليها.

محررو الرسومات ثنائية الأبعاد هم محررون للنمذجة ثنائية الأبعاد، ولكن كما نعلم جميعا فإن النمذجة ثنائية الأبعاد لا تعبر عن الصورة الحقيقية للمناطق الكهربائية. وعلى هذا، فإن الهدف من مشروعنا هذا هو تطوير محرر ثلاثي الأبعاد يُستخدم لإنشاء المخططات الكهربائية ثلاثية الأبعاد، ولتلبية هذه الاحتياجات استخدمنا برامج النمذجة ثلاثية الأبعاد لتوفير المكونات الأساسية، كما قمنا بنقل هذه المكونات إلى جافا اف اكس باستخدام مكتبة متخصصة في ذلك .

تحدثنا في هذه الأطروحة عن الخطوات اللازمة لتحقيق هذا المحرر، كما تطرقنا إلى أبرز الأدوات المستخدمة لتطويره

Abstract

As part of our work, we are interested in the data control, supervision and acquisition system SCADA which is a supervision system which processes a large number of measurements and remotely controls the installations. 2D graphics editors are editors for 2D modeling, but as we all know that 2D modeling does not represent the real picture of the electrical regions.

Therefore, the objective of this study is to develop a 3D graphic editor which is used to build 3D electricity plans. To achieve these needs, we used 3D modeling tools (SketchUp and Blender) to build the necessary components, and we implemented these components in javaFx using the "interactivemesh" library.

In this thesis, we talked about the necessary steps for the realization of this editor, also the tools used during the realization of the editor.

Table des matières

Chapitre I Introduction	1
I. La problématique	2
I.1. Les objectifs	2
II. Généralité sur les systèmes SCADA	3
II.1. Définition	3
II.2. Historique	4
II.3. Les composants d'un système SCADA	7
A. Interface Homme Machine -IHM-	7
B. Système Informatique de Surveillance	7
C. Unité Terminale Distante <i>RTU</i>	7
D. Automate Programmable Industriel (APIs) - <i>Programmable Logic Controllers PLCs</i> -	7
E. L'infrastructure de communication	8
III. Les systèmes SCADA en 3D	9
III.1. Définition de la modélisation 3D	9
III.2. La différence entre un modèle 2D et un modèle 3D	10
III.3. Exemples des systèmes SCADA en 3D	11
A. Proposition de méthode de génération et de personnalisation pour la construction d'une interface graphique 3D pour SCADA	11
III.4. Le système SCADA à Sonelgaz	13
A. Sonelgaz	13
B. ELIT	15
C. Le système SCADA à Sonelgaz	15

Chapitre II	Conception de l'éditeur graphique 3D	16
I.	Comparaison entre notre projet et le projet japonais	17
II.	Introduction au langage UML	17
II.1.	Introduction au Génie Logiciel	17
A.	Modèles de cycle de vie d'un logiciel	19
A.a.	Modèle en Cascade	19
A.b.	Modèle en V	19
A.c.	Modèle en spirale	21
B.	Le cycle de vie exprimé dans notre cas	22
II.2.	Langage UML - <i>Unified Modeling Language</i> -	23
A.	Introduction	23
B.	Définition	23
C.	Les diagrammes UML 2.5	24
C.a.	Les diagrammes de structure	24
C.b.	Les diagrammes de comportement	25
D.	Pourquoi UML?	27
III.	Conception de l'éditeur graphique 3D	28
III.1.	Diagramme de classe	28
III.2.	Diagramme de cas d'utilisation	30
III.3.	Diagrammes de séquence	31
A.	S'authentifier	31
B.	Ouvrir une synoptique	32
C.	Choisir une synoptique	33
Chapitre III	Réalisation de l'éditeur graphique 3D	34
I.	La modélisation 3D	35
I.1.	Définition	35
I.2.	Les avantages	36
I.3.	Les inconvénients	36
I.4.	Les outils de la modélisation 3D	36
I.5.	Les logiciels utilisés dans notre projet	38
A.	SketchUp	38

A.a.	Historique	39
A.b.	Avantages	39
A.c.	Inconvénients	40
A.d.	Comment on a utilisé SketchUp?	41
B.	Blender	41
B.a.	Historique	41
B.b.	Avantages	42
B.c.	Inconvénients	43
B.d.	Comment on a utilisé Blender?	43
I.6.	L'architecture MVC	44
.a.	Le modèle	45
.b.	La vue	45
.c.	Le contrôleur	45
A.	Comment ces éléments communiquent entre eux?	46
B.	Avantages	47
C.	Inconvénients	47
Chapitre IV	Description de l'éditeur réalisé	48
I.	L'interface de l'éditeur	49
I.1.	Introduction	49
I.2.	Le contenu de l'interface graphique de l'éditeur	49
I.3.	Description de l'interface	50
A.	Création du schéma	50
B.	Les informations des composants	51
C.	La connectivité des composants	53
D.	La sauvegarde des schémas 3D	54
E.	Comparaison entre une modélisation 2D et une modélisation avec l'éditeur graphique 3D	54
F.	Les caractéristiques de notre éditeur graphique 3D	56
Conclusion		57

Table des figures

1.1	Un simple système SCADA avec un seul ordinateur	4
1.2	Modèle de données basé sur un ordinateur central au début des années 1970.	5
1.3	Système SCADA multi-niveaux.	6
1.4	Architecture d'un API	8
1.5	Une interface graphique 2D pour 20 nœuds	12
1.6	Une interface graphique 3D pour 20 nœuds	13
1.7	Filiales du groupe Sonelgaz	14
2.1	Modèle de cycle de vie en Cascade	19
2.2	Modèle de cycle de vie en V	21
2.3	Modèle de cycle de vie en spirale	22
2.4	Présentation des diagrammes UML 2.5	26
2.5	Diagramme de classe de l'éditeur graphique 3D	28
2.6	Diagramme de cas d'utilisation de l'éditeur graphique 3D	30
2.7	Diagramme de séquence de la cas d'utilisation "S'authentifier"	31
2.8	Diagramme de séquence de la cas d'utilisation "Ouvrir une synoptique"	32
2.9	Diagramme de séquence de la cas d'utilisation "Choisir une synoptique"	33
3.1	Exemple de disjoncteur édité par SketchUp.	38
3.2	Exemple de transformateur électrique édité par SketchUp.	40
3.3	Exemple d'arc et cercle en SketchUp.	41
3.4	Figure exprime la structure compète d'un composant et la partie retirée.	43
3.5	Figure exprime le découpage du composant précédent en le donnant un identifiant.	44
3.6	L'architecture MVC.	45

3.7	La requête du client arrive au contrôleur et celui-ci lui retourne la vue.	46
4.1	Une capture d'écran de l'interface graphique de l'éditeur 3D.	50
4.2	Une capture d'écran de l'espace où s'affiche notre schéma 3D.	51
4.3	Une capture d'écran de l'espace où s'affiche la structure du composant sélectionné.	51
4.4	Une capture d'écran d'un exemple du composant 3D (Pylône).	52
4.5	Une capture d'écran de l'espace où l'utilisateur doit saisir les informa- tions de chaque composant.	52
4.6	Une capture d'écran qui montre les flèches dirigeantes.	53
4.7	Une capture d'écran qui montre la connectivité entre deux (02) compo- sants.	54
4.8	Un schéma 2D faite par l'ancien éditeur 2D de Elit.	55
4.9	Le schéma 2D précédent faite par notre éditeur 3D.	55

Liste des tableaux

1.1	Comparaison entre un modèle 2D et un modèle 3D	10
-----	--	----

CHAPITRE

I

INTRODUCTION

I. La problématique

La production, le transport et la distribution de l'électricité du groupe SONELGAZ utilise un système de contrôle, de supervision et d'acquisition des données en temps réel « SCADA » pour le suivi et le pilotage de son immense réseau.

Ce système abrite plusieurs serveurs tel que le serveur d'ingénierie qui est un éditeur graphique 2D, qui consiste à créer des diagrammes et des schémas synoptiques en 2D avec des composants et des symboles prédéfinis. Chaque composant doit être associé à une adresse pour représenter un point sur le réseau, et tout ceci sera exporter en format XML vers les autres serveurs pour les exploiter.

D'après la comparaison faite dans les titres précédents, on a vu que la modélisation 3D a des propriétés très importantes et qui n'existe plus dans la modélisation 2D, tel que la possibilité d'être visualisée sous n'importe quel angle de l'écran, et aussi elle nous montre clairement l'emplacement et la taille de chaque équipement. Par conséquence on a conclu que malgré les points faibles de la modélisation 3D mais aussi par contre elle donne un effet très réaliste, proche de l'espace dans lequel nous vivons en comparant avec la modélisation 2D.

D'après ce qui est passé et dans le cadre de la montée en version de la solution SCADA de Sonelgaz - développée par moyens propre - l'objectif de notre travail est la réalisation d'un éditeur graphique 3D, dans le but de schématiser la synoptique du réseau électrique en 3D au niveau du serveur d'ingénierie.

I.1. Les objectifs

Le projet consiste à développer un logiciel qui permet d'illustrer des schémas graphiques en 3D représentant une infrastructure plus réaliste du réseau électrique et la topologie du système de contrôle et de supervision schéma unifilaire, réseau LAN, réseau de communication.

II. Généralité sur les systèmes SCADA

II.1. Définition

Un système SCADA *Supervisory Control And Data Acquisition* -En français- contrôle de surveillance et acquisition de données, est un système de contrôle d'automatisation utilisé dans des secteurs tels que l'électricité, le gaz, l'énergie, le pétrole, l'eau et bien d'autres encore. Il s'agit d'un système centralisé qui surveille et contrôle des sites entiers, allant d'une usine industrielle à un complexe d'usines dans tout le pays, exemple : Sonelgaz. Un système SCADA collecte les données de divers capteurs dans une usine, ou dans d'autres lieux éloignés et envoie ensuite ces données à un ordinateur central qui les gère et les contrôle.[1]

Un système SCADA fonctionne en utilisant des signaux qui communiquent par des canaux pour fournir à l'utilisateur des commandes à distance de tout équipement d'un système donné. Il met également en œuvre une base de données distribuée, ou base de données d'étiquettes, qui contient des étiquettes ou des points dans toute l'usine. Ces points représentent une seule valeur d'entrée ou de sortie qui est surveillée ou contrôlée par le système SCADA dans la salle de contrôle centralisée.

Un système SCADA se compose de quatre (04) serveurs :

- **Serveur d'ingénierie** : Dans ce serveur la conception et l'association des points (création des schémas) sera faite.
- **Serveur de communication** : Ce serveur traduit les informations reçues à partir de l'unité terminale distante -RTU-.
- **Serveur d'application** : Ce qui affiche les données reçues du serveur Ingénierie, et faire une connexion bidirectionnelle (Requête/Réponse) avec le serveur de communication.
- **Serveur d'archivage** : Ce serveur enregistre les données venues du serveur d'application en utilisant un *Buffer*.

Le système SCADA et les autres systèmes de contrôle sont très importants car ils contrôlent la plupart de nos produits. Les communications SCADA conventionnelles

ont été des communications en série point à multipoint sur des lignes de location ou des systèmes radio privés. Avec l'avènement du protocole Internet (IP), la technologie IP a vu son utilisation augmenter dans les communications SCADA. [2]

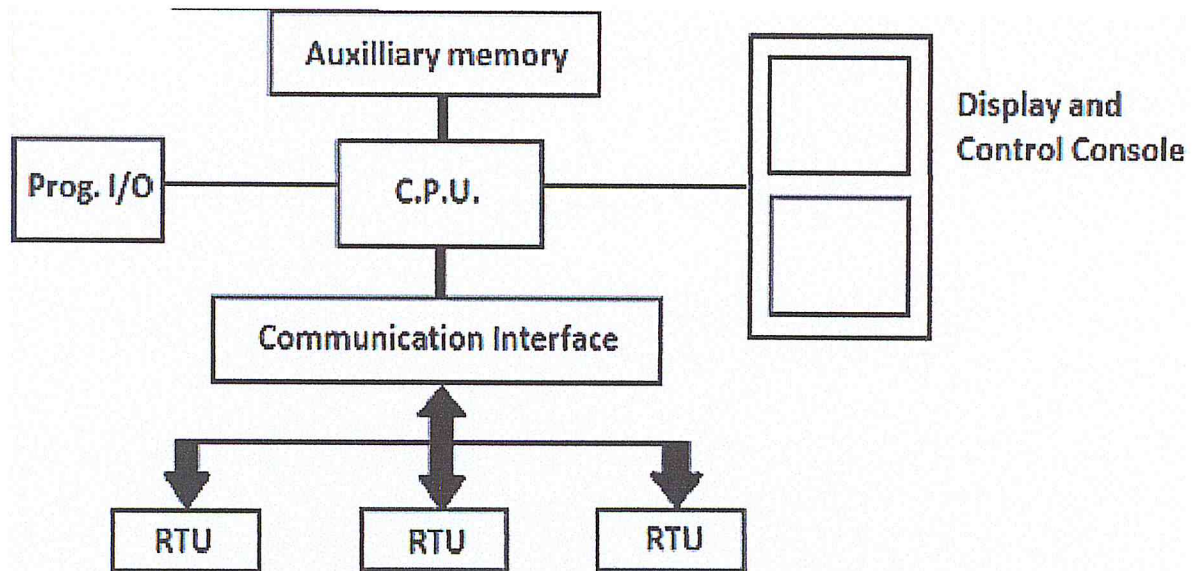


FIGURE 1.1: Un simple système SCADA avec un seul ordinateur

II.2. Historique

En 1960, la première fois que le système SCADA a été introduit, le premier système SCADA réalisé avec des mini-ordinateurs qui chronomètrent les services de réseau normaux non disponibles, l'histoire des systèmes de contrôle est ancrée dans les premiers systèmes de télémétrie et de chemin de fer des fusées. Les essais de fusée nécessitaient une méthode pour collecter des données et les transmettre au sol avant que la fusée n'explose. Dans les chemins de fer, les bureaux centraux utilisaient des communications filaires pour surveiller la position des trains et l'état des aiguillages.

Au début des années 1970, les systèmes SCADA ont commencé avec un grand ordinateur de type mainframe unique. Les entrées ont été faites à partir de plusieurs productions des machines sur des terminaux avec des écrans à base de texte.

Ces terminaux étaient utilisés par le personnel chargé de faire fonctionner les machines. Cette large gamme de production et les informations commerciales ont été saisies dans des terminaux contrôlé par l'ordinateur unique.

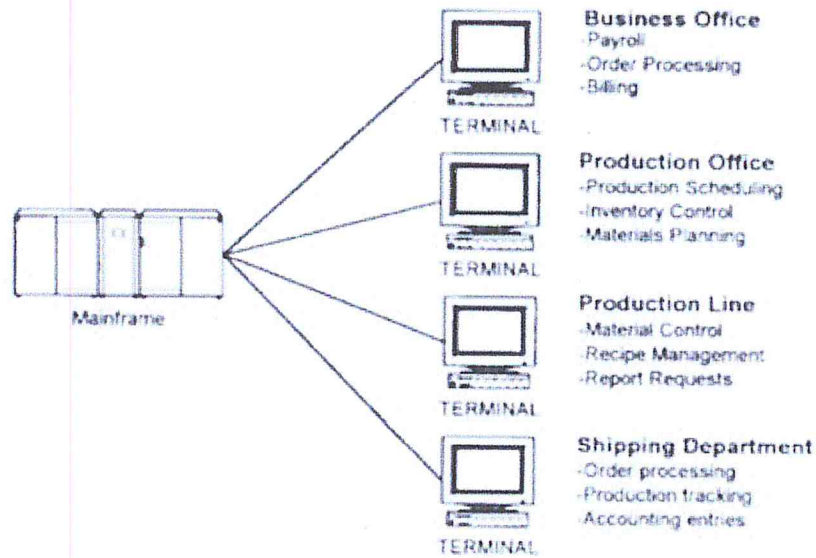


FIGURE 1.2: Modèle de données basé sur un ordinateur central au début des années 1970.

L'avantage de l'architecture de l'ordinateur central est que tous les données sont stockées dans un seul endroit, ce qui facilite l'accès aux données de chaque niveau.

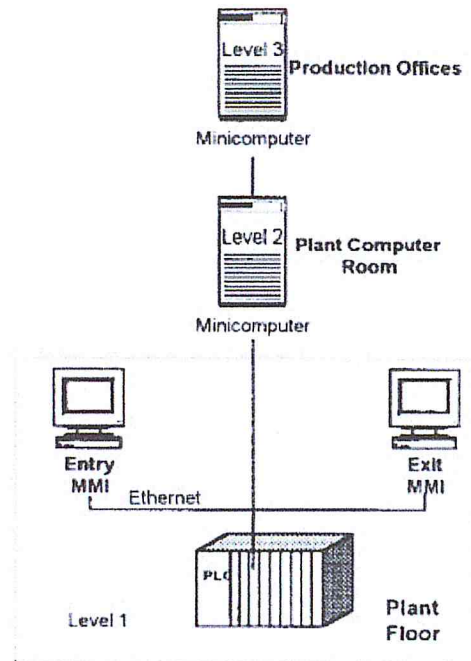


FIGURE 1.3: Système SCADA multi-niveaux.

Toutefois, les inconvénients l'emportent de loin sur les avantages. Ces inconvénients sont :

- Toutes les données doivent être saisies manuellement.
- Il n'y a pas d'entrées et de sorties (E/S) à la machine.
- La flexibilité et l'extensibilité du système étaient très limitées tel que tout changement nécessitait généralement une programmation des modifications ou même des mises à niveau matérielles coûteuses.

Au fur et à mesure des progrès technologiques, les méthodes de communication sont passées de la radio, aux lignes téléphoniques louées et aux modems, aux technologies de mise en réseau. Aujourd'hui, les systèmes de contrôle modernes utilisent souvent le TCP/IP pour les communications entre appareils, principalement pour tirer parti de la flexibilité et du coût des technologies LAN et WAN de base.[3]

II.3. Les composants d'un système SCADA

A. Interface Homme Machine -IHM-

Il fait parti du serveur d'application. Il s'agit d'une interface qui présente les données du processus à un opérateur humain, et par ce biais, l'opérateur humain surveille et contrôle le processus.

B. Système Informatique de Surveillance

Ce système rassemble les informations et les données et envoie les commandes ou les contrôles au processus.

C. Unité Terminale Distante *RTU*

Le système SCADA comporte une unité terminale principale (*Master Terminal Unit*), que l'on pourrait appeler le cerveau du système, et une ou plusieurs unités terminales distantes *RTU*. Les *RTU* rassemblent les données localement et les envoient à la *MTU* qui émet les commandes appropriées à exécuter sur place. Un système de logiciels standard ou personnalisés est utilisé pour rassembler, interpréter et gérer les données.[4]

Autrement dit, Il reçoit les données à partir du système de protection et les envoie au système de supervision SCADA.

D. Automate Programmable Industriel (APIs) -*Programmable Logic Controllers PLCs*-

L'automate Programmable Industriel API est un ordinateur à usage spécifique, dont il n'a pas d'écran, pas de clavier, pas d'imprimante, pas de disque dur, utilisé pour contrôler les systèmes automatisés dans les industries. Il constitue l'une des formes les plus avancées et les plus simples de systèmes de contrôle qui remplacent aujourd'hui à grande échelle les relais logiques câblés qui consomment beaucoup d'électricité, produisent beaucoup de chaleur et de suie, et prennent beaucoup de place.

Les Automates Programmables Industriels a plusieurs avantages tels que :

- L'évolutivité : Ils sont très favorables a l'évolution.

- La taille des applications : Gamme importante d'automate.
- La vitesse : Temps de cycle de quelque *ms*.
- Le développement d'une application et documentation : Très facile avec des outils de programmation de plus en plus puissant.
- L'architecture de commande : Centralisée ou décentralisée avec l'apparition d'une offre importante en choix de réseaux, bus de terrain et blocs E/S déportés.
- La mise en œuvre : Mise au point rendu plus facile avec l'apparition des outils de simulation. [5]

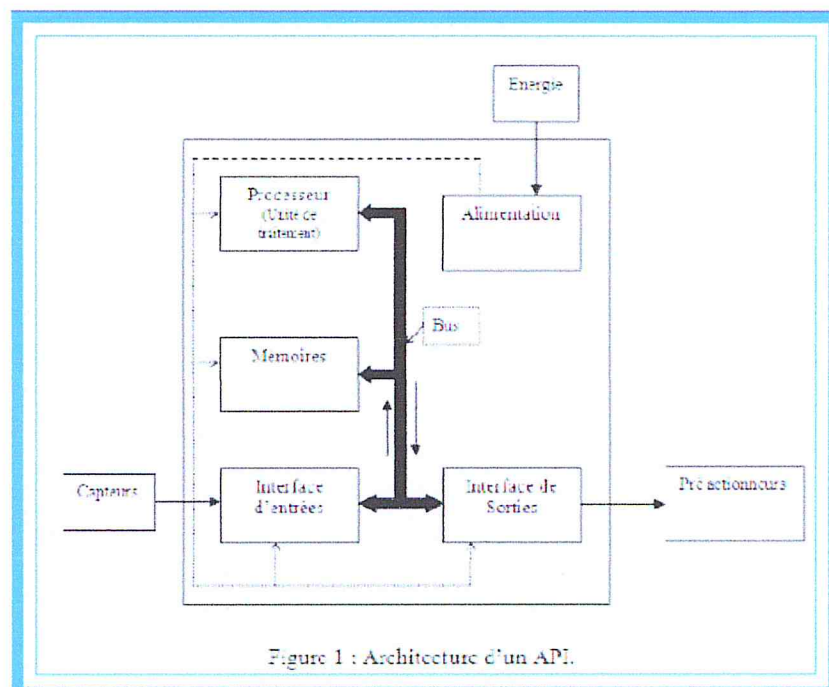


FIGURE 1.4: Architecture d'un API

E. L'infrastructure de communication

Il permet de connecter le système de surveillance aux unités terminales distantes -RTU-.

III. Les systèmes SCADA en 3D

III.1. Définition de la modélisation 3D

La modélisation 3D est une technique infographique récente qui fait partie du domaine informatique, et qui vise à créer - dans un logiciel adapté - des objets tridimensionnels, que ça soit à partir des objets réels ou pas. Elle est utilisée notamment dans le développement des jeux vidéo, des films d'animation, ou encore des plans ou projets d'ordre professionnel, exp : la reconstitution 3D de bâtiments afin de les construire.

Un logiciel de modélisation 3D génère un modèle 3D à l'aide de différents outils et approches, notamment :

- polygones simples.
- primitives 3D : des formes polygonales simples - pyramides, cubes, sphères, cylindres et cônes.
- courbes splines.
- NURBS (B-splines rationnelles non uniformes) : Ce sont des formes douces définies par des lignes incurvées qui sont relativement complexes à traiter informatiquement.

III.2. La différence entre un modèle 2D et un modèle 3D

Un modèle 2D	Un modèle 3D
Les objets bidimensionnels sont créés dans deux axes X et Y	Les objets tridimensionnels sont créés dans trois axes X et Y et Z
Les objets bidimensionnels ne peuvent pas être visualisés sous n'importe quel angle sur un écran plat	Les objets tridimensionnels peuvent être visualisés sous n'importe quel angle sur un écran plat
Il est moins cher que l'éditeur graphique en 3D	Nécessite un coût plus élevé que celui d'un système d'interface utilisateur graphique en 2D
la scène bidimensionnelle ne peut pas montrer le fonctionnement en temps réel des équipements	Observer les conditions de fonctionnement en temps réel des équipements
Ne montre pas la surface de l'équipement	Montre la position et la taille de l'équipement

TABLE 1.1: Comparaison entre un modèle 2D et un modèle 3D

La table au dessus nous montre la différence entre les deux modèles (2D et 3D), et nous montre aussi les avantages et les inconvénients des deux.

D'après cette table, on peut conclure que le modèle 2D est préférable par rapport au modèle 3D si on prend en considération le coût, par contre le modèle 3D est préférable si on prend en considération d'autres mesures, parcequ'il nous montre la surface réelle de l'équipement qui a la possibilité d'être visualisé sur n'importe quel angle sur un écran plat, ce qui n'existe plus dans le modèle 2D qui n'a pas la possibilité d'être visualisé sur des différents angles car il est montré comme un écran plat.

Aujourd'hui, le Groupe Sonelgaz est composé de 16 sociétés directement pilotées par la Holding, de 18 sociétés en participation avec des entités du Groupe et de 10 sociétés en participation avec des tiers.

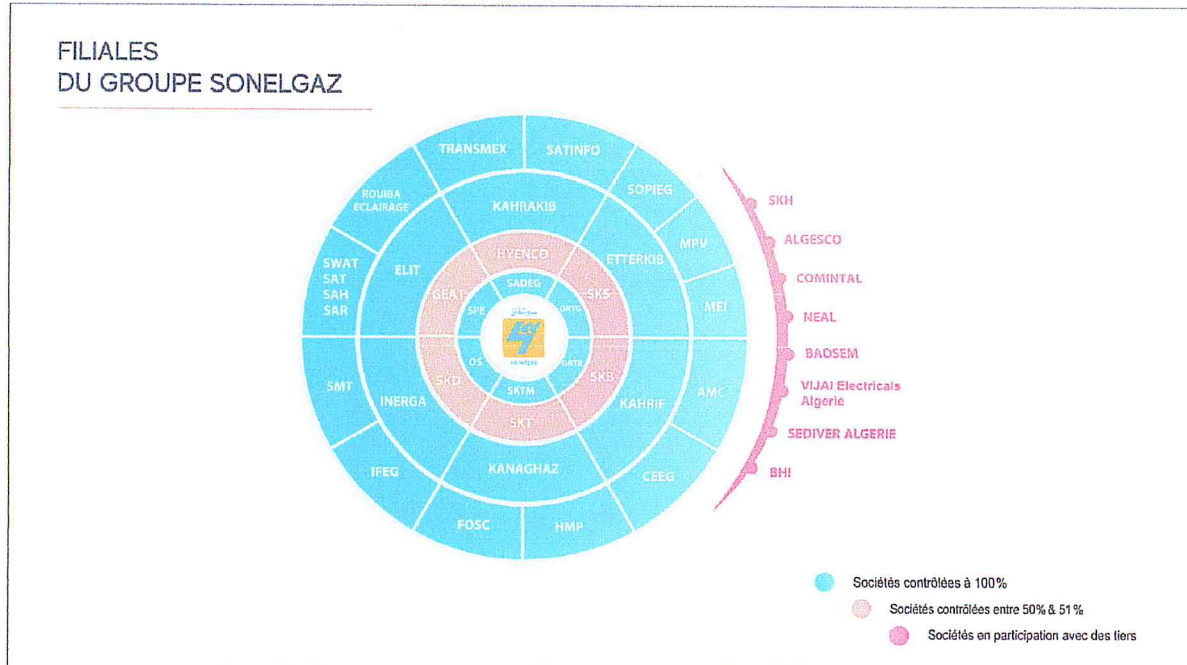


FIGURE 1.7: Filiales du groupe Sonelgaz

Ses filiales métiers de base assurent la production **SPE**, le transport **GRTE** et la distribution de l'électricité **SADEG**, ainsi que le transport **GRTG** et la distribution **SADEG** du gaz par canalisations. Ses filiales travaux sont en charge de la réalisation des infrastructures électriques et gazières du pays. Ses filiales de prestations de service activent principalement dans les domaines de la fabrication et de la maintenance d'équipements énergétiques, la distribution de matériel électrique et gazier, le transport et la manutention exceptionnels.

Les activités opérationnelles étant dévolues à ses sociétés, la holding Sonelgaz assure le pilotage du Groupe. A ce titre, elle élabore et met en œuvre la stratégie de développement du Groupe dans son ensemble, ainsi que la politique financière et Ressources Humaines. Aussi, elle œuvre à mobiliser des financements importants afin de développer et renforcer l'infrastructure électrique et gazière. Le marché africain en plein essor lui offre l'occasion d'exporter le savoir-faire de ses filiales, notamment au Mali, Libye, Mauritanie et Soudan...[7]

B. ELIT

Spécialisée dans les technologies de l'information et de la communication, *El Djazair Information Technology "ELIT"* est une société algérienne comptant plus de 300 ingénieurs informaticiens, plus de 40 clients et des infrastructures hautement disponibles et sécurisées.

Au-delà des aspects reconnus au domaine IT, les réseaux informatiques, le développement des sites web, la messagerie électronique, etc., ELIT assure la sécurité des systèmes d'information via une plateforme de sécurité à la pointe de la technologie et ce, avec une ressource humaine 100% algérienne.[8]

C. Le système SCADA à Sonelgaz

Le système SCADA, ou station informatique de télésurveillance à distance, permet de recevoir des alarmes instantanées, sonores, visuelles et en transcription sur écran, à travers le réseau électrique de tout le territoire de la wilaya, ont expliqué les responsables et les ingénieurs de la SDC Sonelgaz. Ce système, qui fonctionne avec des supports aériens télécommandés placés sur des pylônes-relais, hauts de 45 mètres, implantés dans plusieurs sites de la wilaya pour la transmission par ondes comme ceux de la téléphonie mobile, permet non seulement de localiser avec précision toute panne, coupure de courant électrique, baisse de tension, agression (vol de câbles par exemple), mais d'établir aussi les lignes et en tout lieu du réseau.[9]



I. Comparaison entre notre projet et le projet japonais

Le projet japonais intitulé par " Proposition de méthode de génération et de personnalisation pour la construction d'une interface graphique 3D pour SCADA " qui est défini déjà dans le premier chapitre (Introduction) s'appuie sur une interface graphique 2D qui existe déjà, et à la base de cette interface ils ont créé une bibliothèque *GhostHouse* tel que chaque composant appartenant à cette bibliothèque correspond à un composant de l'interface graphique 2D qui existe déjà, sachant que cette bibliothèque est faite sans le passage par la programmation.

Par contre, notre travail se base surtout sur des composants 3D existe déjà dans la grande bibliothèque 3D de *SketchUp* appelée *3D Warehouse*, le travail à faire dans notre cas est la construction d'une bibliothèque contenant un nombre limité des composants électrique 3D. Parfois, les composants qu'on a retiré de *3D Warehouse* seront intégrés tel qu'ils sont, mais parfois on est besoin de prendre juste une partie de ce composant et non pas toute la structure du composant, ce qu'on peut faire en utilisant le logiciel appelé *Blender*.

II. Introduction au langage UML

II.1. Introduction au Génie Logiciel

Un logiciel ou une application est un ensemble de programmes qui permet à un ordinateur ou à un système informatique d'assurer une tâche ou une fonction en particulier, exemple : logiciel de comptabilité.

Les logiciels, suivant leur taille, 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 d'importants problèmes de conception et de coordination. Or, le développement d'un logiciel est une phase absolument cruciale qui monopolise l'essentiel du coût d'un produit et conditionne sa réussite et sa pérennité.

Le génie logiciel est un domaine de recherche qui a été défini la fin des années 60, à Garmisch-Partenkirchen en Allemagne.

Le génie logiciel est un ensemble des méthodes, des techniques et des outils dédiés à

la conception, au développement et à la maintenance des systèmes informatiques qui a pour objectif d'avoir des procédures systématiques pour des logiciels de grande taille afin que :

- La spécification corresponde aux besoins réels du client.
- le logiciel respecte sa spécification.
- Les délais et les coûts alloués à la réalisation soient respectés.

Pour savoir la qualité d'un logiciel il faut que notre logiciel avoir un ensemble de critères. Parmi ces derniers nous pouvons citer :

- **Validité** : Aptitude d'un produit logiciel à remplir exactement ses fonctions, définies par le cahier des charges et les spécifications.
- **Fiabilité** : Aptitude d'un produit logiciel à fonctionner dans des conditions anormales.
- **Compatibilité** : Veut dire que le logiciel peut être combiner avec d'autre logiciels.
- **Extensibilité** : Autrement dit : Maintenance, que veut dire la facilité avec laquelle un logiciel se prête à sa maintenance, c'est-à-dire à une modification ou à une extension des fonctions qui lui sont demandées.
- **Efficacité** : C'est de l'utilisation optimale des ressources matérielles.
- **Intégrité** : Veut dire la protection du code et des données contre les accès non autorisés.
- **Facilité d'utilisation** : Facilité d'apprentissage, d'utilisation, de préparation des données, d'interprétation des erreurs et de rattrapage en cas d'erreur d'utilisation.

Remarque : Ces critères et facteurs ne sont pas toujours les mêmes, parce qu'ils sont parfois contadicoires, donc le choix doit s'effectuer en fonction du contexte.[10]

A. Modèles de cycle de vie d'un logiciel

A.a. Modèle en Cascade

Le modèle de cycle de vie en cascade a été mis au point dès 1966, puis formalisé aux alentours de 1970.

Il est un modèle de gestion linéaire.

Parmi les caractéristiques de ce modèle :

- Chaque étape doit être terminée avant que la suivante commence.
- A chaque étape, une production d'un documents base de l'étape de suivante.
- La découverte d'une erreur entraîne un retour à la phase d'origine de l'erreur et nouvelle cascade, avec de nouveaux documents.

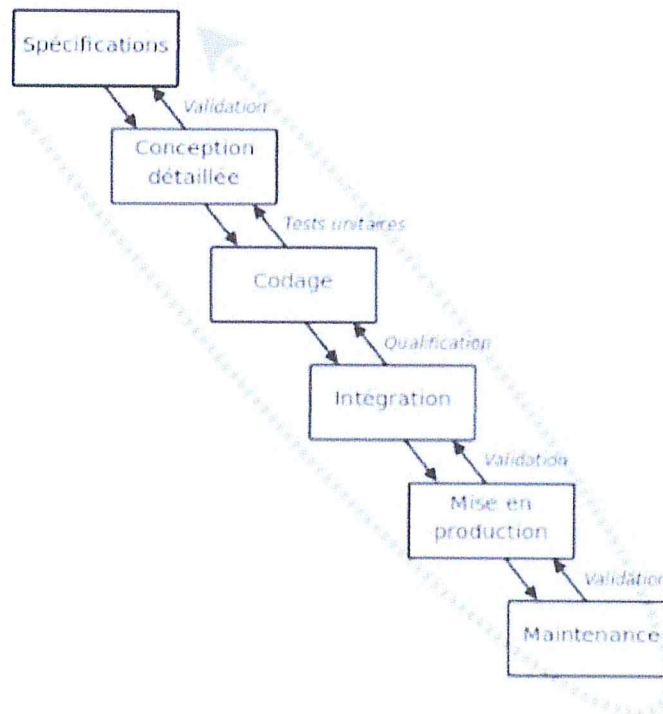


FIGURE 2.1: Modèle de cycle de vie en Cascade

A.b. Modèle en V

Le modèle en V demeure actuellement le cycle de vie le plus connu et certainement le plus utilisé. Il s'agit d'un modèle en cascade dans lequel le développement des tests et du logiciels sont effectués de manière synchrone.

Le cycle en V est un cycle composé de 3 grandes phases contenant 8 étapes de conception d'un produit :

- **La phase de conception** : Cette phase contient :
 1. Expressions des besoins.
 2. Spécifications.
 3. Conception préliminaire.
 4. Conception détaillée.
- **La phase de réalisation** : Cette phase contient :
 1. Codage.
 2. Tests unitaires.
- **La phase de validation** : Cette phase contient :
 1. Tests d'intégration.
 2. Tests de validation.
 3. Utilisation.

Le modèle a plusieurs caractéristiques qui sont les suivantes :

- Chaque phase ne commence qu'à partir du moment où la précédente est terminée.
- Mise en évidence de la complémentarité des phases menant à la réalisation et des phases de test permettant de les valider.
- En face de chaque phase de spécification, il est mis en place un système de vérification qui assure un meilleur produit au final.
- Possède la particularité de positionner en parallèle des activités de contrôle (Cette caractéristique est une implication de la précédente).

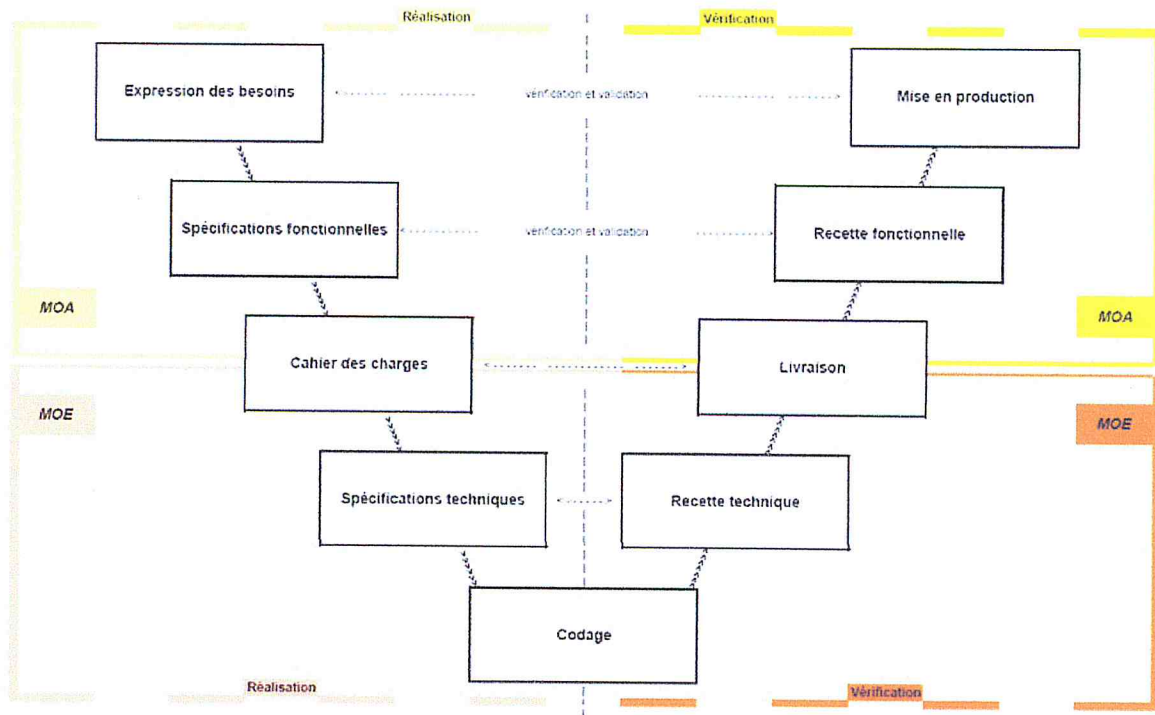


FIGURE 2.2: Modèle de cycle de vie en V

A.c. Modèle en spirale

Ce modèle a été défini par Barry Boehm en 1988, il est représenté à l'aide d'une spirale, tel que chaque boucle représente une phase du développement, et la boucle la plus interne traite des premières phases (faisabilité) et la plus externe traite de la livraison.

Chaque boucle traverse quatre sections :

- Définition des objectifs de la phase.
- Evaluation des risques.
- Développement et validation.
- Planification de la phase suivante.

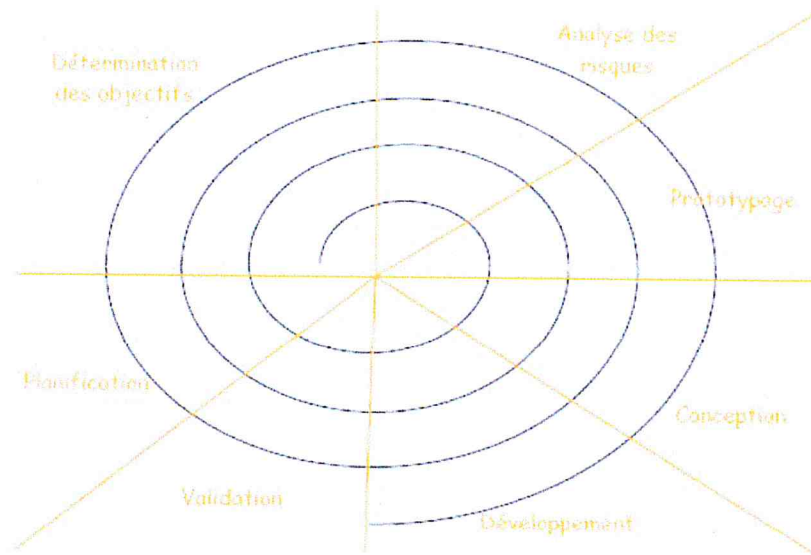


FIGURE 2.3: Modèle de cycle de vie en spirale

B. Le cycle de vie exprimé dans notre cas

En prenant en compte les avantages et les inconvénients de chaque cycle de vie des projets et en prenant en compte aussi notre cas, le cycle de vie en V est le plus adaptable, à cause des parcequ'il est divisé en trois (03) grandes phases en contenant huit (08) étapes de conception, et aussi à cause des tests faits à la fin de chaque phase, ce que peut faciliter le travail à partir de la bonne organisation de ce type de cycle de vie.

Le cycle de vie en V est le plus connu et certainement le plus utilisé. Contrairement au modèle en cascade, le modèle en V fait apparaitre le fait que le début du processus de développement conditionne ses dernières étapes.

II.2. Langage UML -*Unified Modeling Language*-

A. Introduction

Les méthodes utilisées dans les années 1980 pour organiser la programmation impérative (notamment Merise) étaient fondées sur la modélisation séparée des données et des traitements. Lorsque la programmation par objets prend de l'importance au début des années 1990, la nécessité d'une méthode qui lui soit adaptée devient évidente. Plus de cinquante méthodes apparaissent entre 1990 et 1995 (Booch, Classe-Relation, Fusion, HOOD, OMT, OOA, OOD, OOM, OOSE, etc.) mais aucune ne parvient à s'imposer. En 1994, le consensus se fait autour de trois méthodes :

1. OMT - *Object Modeling Technic* - de James Rumbaugh (General Electric) fournit une représentation graphique des aspects statique, dynamique et fonctionnel d'un système.
2. OOD - *Object Oriented Design* - de Grady Booch, définie pour le Department of Defense, introduit le concept de paquetage (package).
3. OOSE - *Object Oriented Software Engineering* - d'Ivar Jacobson (Ericsson) fonde l'analyse sur la description des besoins des utilisateurs (cas d'utilisation, ou use cases).

UML hérite principalement de ces trois méthodes (OMT, OOD et OOSE) dont le développement a débuté en 1994, et a été standardisé en 1997 par l'OMG -*Object Management Group*- dans sa version 1.1 qui s'appuie essentiellement sur la technologie objet et les concepts qu'elle véhicule.

Actuellement, UML est dans la version 2.5.

B. Définition

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue.

UML propose un ensemble de notations pour que chacun ait à sa disposition les éléments nécessaires à la conception d'une application.

En résumé, le langage UML est un langage de modélisation qui a pour but de fa-

cilité les transitions, lors du développement d'un projet, du besoin originel à la phase d'implémentation.

Autrement dit, le langage UML est un langage graphique pour visualiser, spécifier, construire et documenter un logiciel.

C. Les diagrammes UML 2.5

Il est impossible de donner une représentation graphique complète d'un logiciel, ou de tout autre système complexe, de même qu'il est impossible de représenter entièrement une statue (à trois dimensions) par des photographies (à deux dimensions). Mais il est possible de donner sur un tel système des vues partielles, analogues chacune à une photographie d'une statue, et dont la conjonction donnera une idée utilisable en pratique sans risque d'erreur grave.

UML 2.5 comporte quinze types de diagrammes représentant autant de vues distinctes pour représenter des concepts particuliers du système d'information. Ils se répartissent en deux grands groupes : **Les diagrammes de structure** - *UML structure* - et **les diagramme de comportement** - *UML behavior* -.

C.a. Les diagrammes de structure

Les diagrammes de structure montrent la structure statique du système et de ses parties à différents niveaux d'abstraction et de mise en œuvre et comment ces parties sont liées les unes aux autres.

Les diagrammes de structure n'utilisent pas de concepts liés au temps, ne montrent pas les détails du comportement dynamique. Cependant, ils peuvent montrer des relations avec les comportements des classificateurs présentés dans les diagrammes de structure.

1. **Diagramme de classe** : Affiche la structure du système, sous-système ou composant conçu en tant que classes et interfaces associées, avec leurs caractéristiques, contraintes et relations - associations, généralisations, dépendances, etc.
2. **Diagramme d'objets** : Diagramme de classe au niveau de l'instance qui montre les spécifications d'instance des classes et des interfaces (objets), des emplacements avec des spécifications de valeur et des liens (instances d'association).

La spécification UML 2.5 ne fournit simplement aucune définition du diagramme d'objet.

3. **Diagramme de paquets** : Affiche les packages et les relations entre les packages.
4. **Diagramme de structure composite** : Ce diagramme pourrait être utilisé pour montrer la structure interne d'un classifieur et le comportement de collaboration.
5. **Diagramme des composants** : Affiche les composants et les dépendances entre eux.
6. **Diagramme de déploiement** : Affiche l'architecture du système en tant que déploiement (distribution) d'artefacts logiciels sur les cibles de déploiement.
7. **Diagramme de profile** : Spécialisation et personnalisation pour un domaine particulier d'un meta-modèle de référence d'UML.

C.b. Les diagrammes de comportement

Les diagrammes de comportement montrent le comportement dynamique des objets dans un système, qui peut être décrit comme une série de modifications du système au fil du temps.

1. **Diagramme des cas d'utilisation** : Représentation des possibilités d'interaction entre le système et les acteurs.
2. **Diagramme d'activités** : Affiche la séquence et les conditions de coordination des comportements de niveau inférieur, plutôt que les classificateurs qui possèdent ces comportements.
3. **Diagramme de la machine d'état (Etats transition)** : Utilisé pour modéliser un comportement discret via des transitions d'états finis.
4. **Diagramme d'interaction** : Ils comprennent plusieurs types de diagrammes différents, qui sont : Diagramme de séquence, diagramme de communication, chronogramme, et diagramme de vue d'ensemble d'interaction.
 - 4.1. **Diagramme de Séquence** : Se concentre sur l'échange des messages entre les lignes de vie (objets).
 - 4.2. **Diagramme de communication** : Se concentre sur l'interaction entre les lignes de vie où l'architecture de la structure interne et la façon dont cela correspond

au message est central.

- 4.3. **Diagramme temporel - Chronogramme** - : Se concentre sur les conditions changeant à l'intérieur des lignes de vie et entre elles le long d'un axe temporel linéaire.
- 4.4. **Diagramme de vie d'ensemble de l'interaction** : Se concentre sur la vue d'ensemble du flux de contrôle où les nœuds sont des interactions ou des utilisations d'interaction.[11]

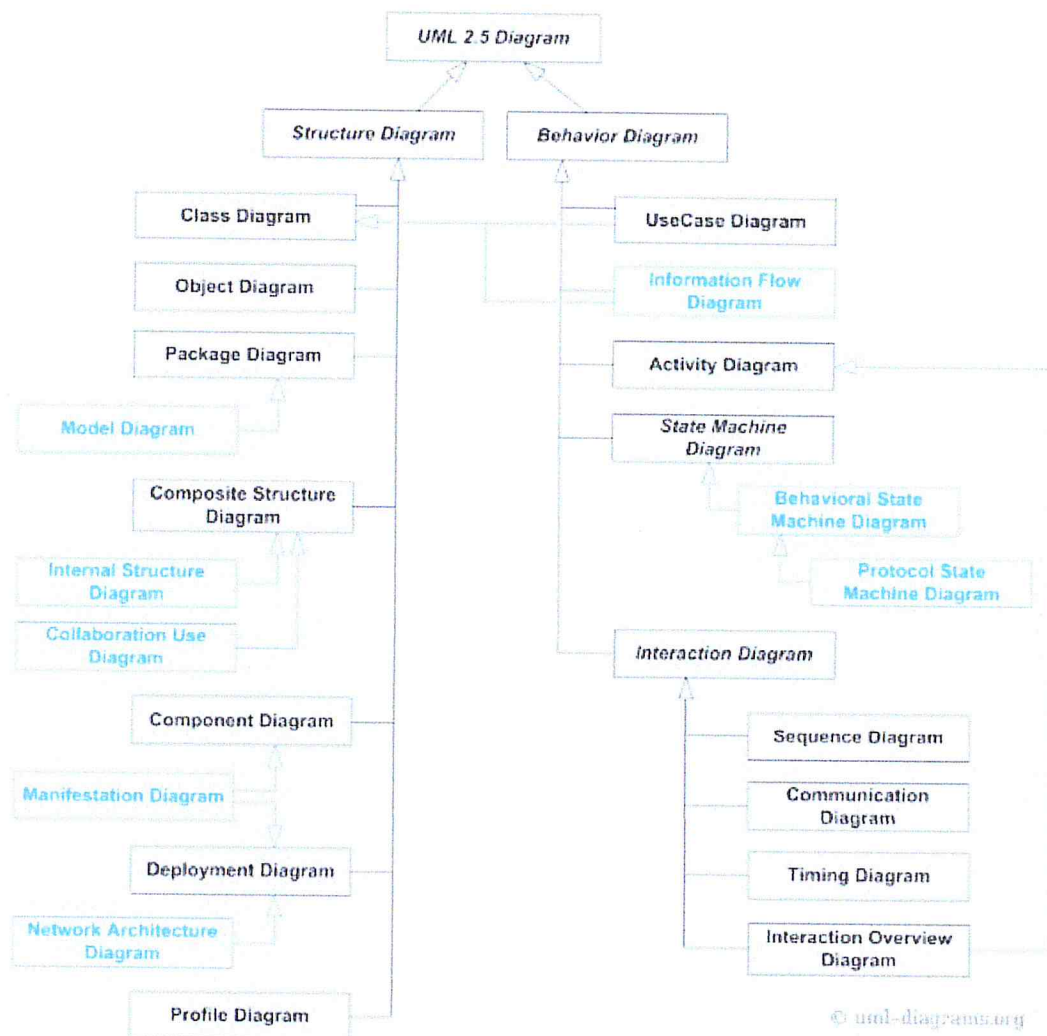


FIGURE 2.4: Présentation des diagrammes UML 2.5

Remarque : Les éléments en bleu ne font pas partie de la taxonomie officielle des diagrammes UML 2.5.

D. Pourquoi UML ?

Le langage UML est une synthèse de tous les concepts et formalismes méthodologiques les plus utilisés, pouvant être utilisé, grâce à sa simplicité et à son universalité, comme langage de modélisation pour la plupart des systèmes devant être développés.

Le langage UML permet ainsi d'apporter des solutions lors du développement de systèmes informatisés :

- Décompresser le processus de développement en distinguant la phase d'analyse (aspects fonctionnel) de la phase de réalisation (aspects technologiques et architecturaux).
- Décompresser le système en sous-systèmes plus facilement abordables : Réduction de la complexité, répartition du travail, réutilisation des sous-systèmes.
- Utiliser une technologie de haut niveau proche de la réalité pour aborder le développement.

III. Conception de l'éditeur graphique 3D

III.1. Diagramme de classe

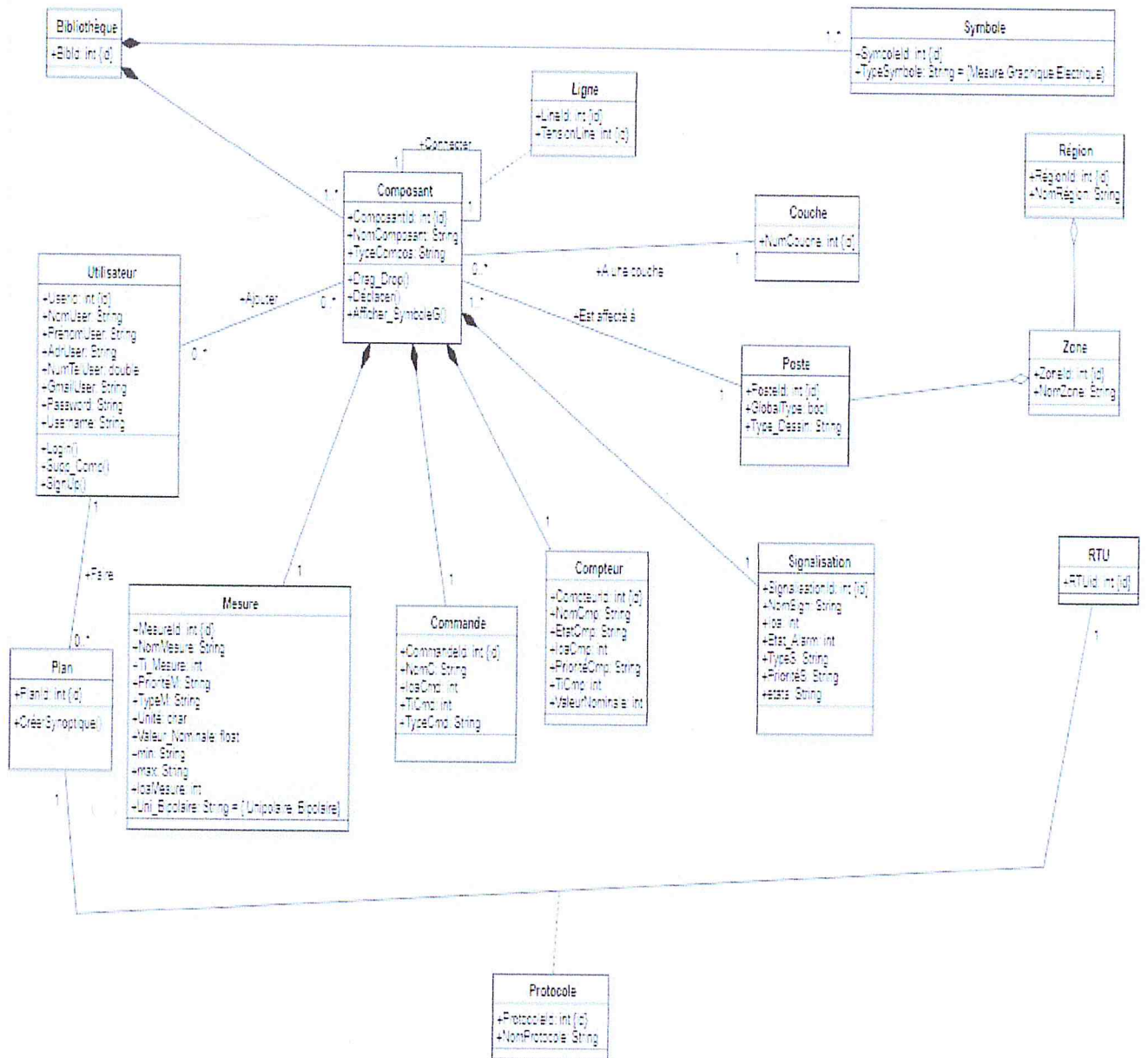


FIGURE 2.5: Diagramme de classe de l'éditeur graphique 3D

Le diagramme de classes contient une classe qui est comme " l'axe " du diagramme qui est la classe 'Composant', cette classe appartient à une bibliothèque qui contient aussi un ensemble de symboles. Chaque composant a un ensemble d'informations (Mesure, Commande, Compteur, Signalisation) et aussi a une couche qui veut dire dans quel niveau de zoom le composant soit visible. Chaque composant aussi appartient à un poste, sachant que la région est un ensemble de zones et la zone est un ensemble de poste.

L'utilisateur crée un plan et lui ajoute des composants, ces composants peuvent être connectés entre eux par des lignes qui ont une tension.

Tous ce qui précède fait parti du serveur d'ingénierie (Le plan) qui communique avec le RTU par des protocoles.

III.2. Diagramme de cas d'utilisation

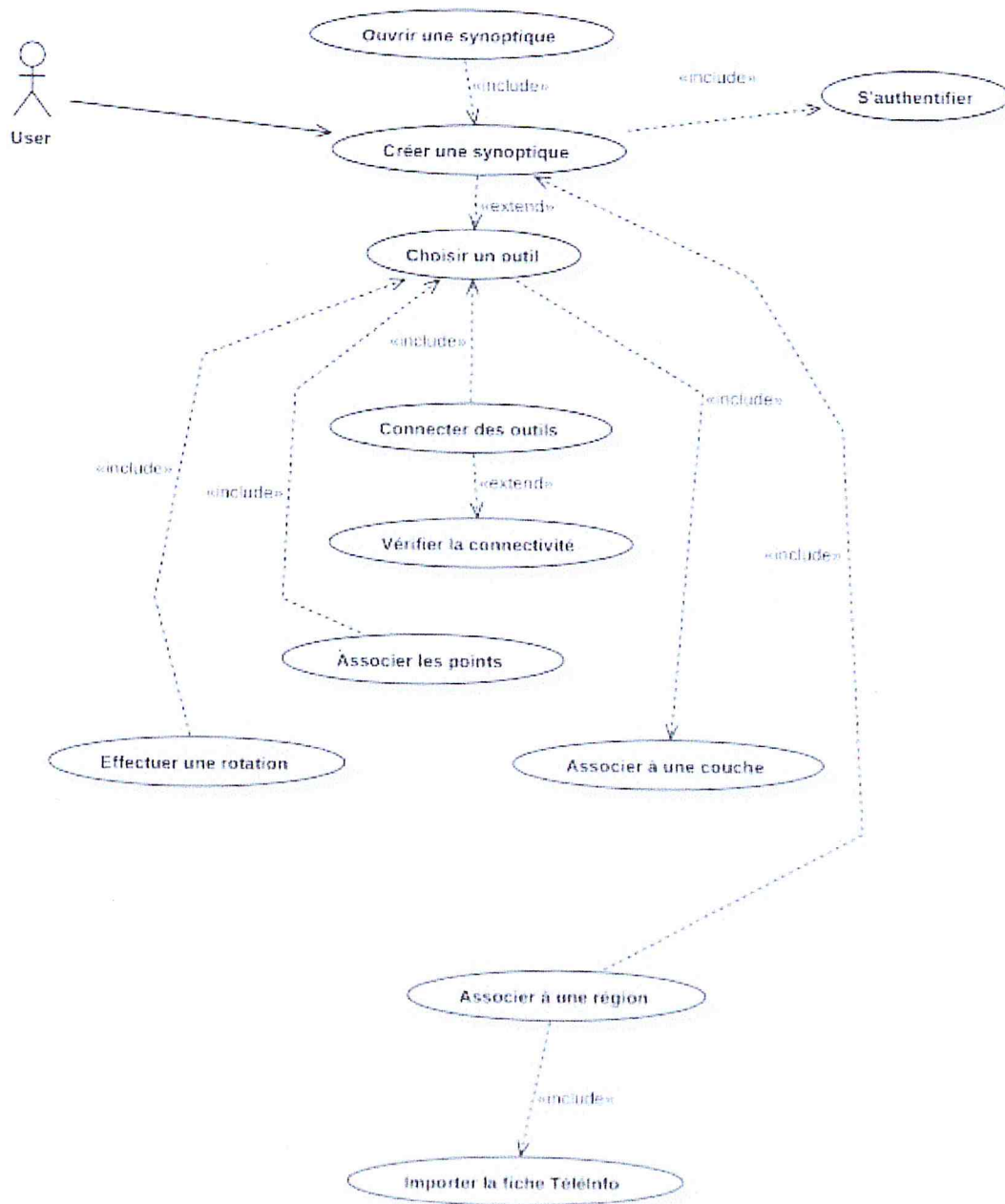


FIGURE 2.6: Diagramme de cas d'utilisation de l'éditeur graphique 3D

III.3. Diagrammes de séquence

A. S'authentifier

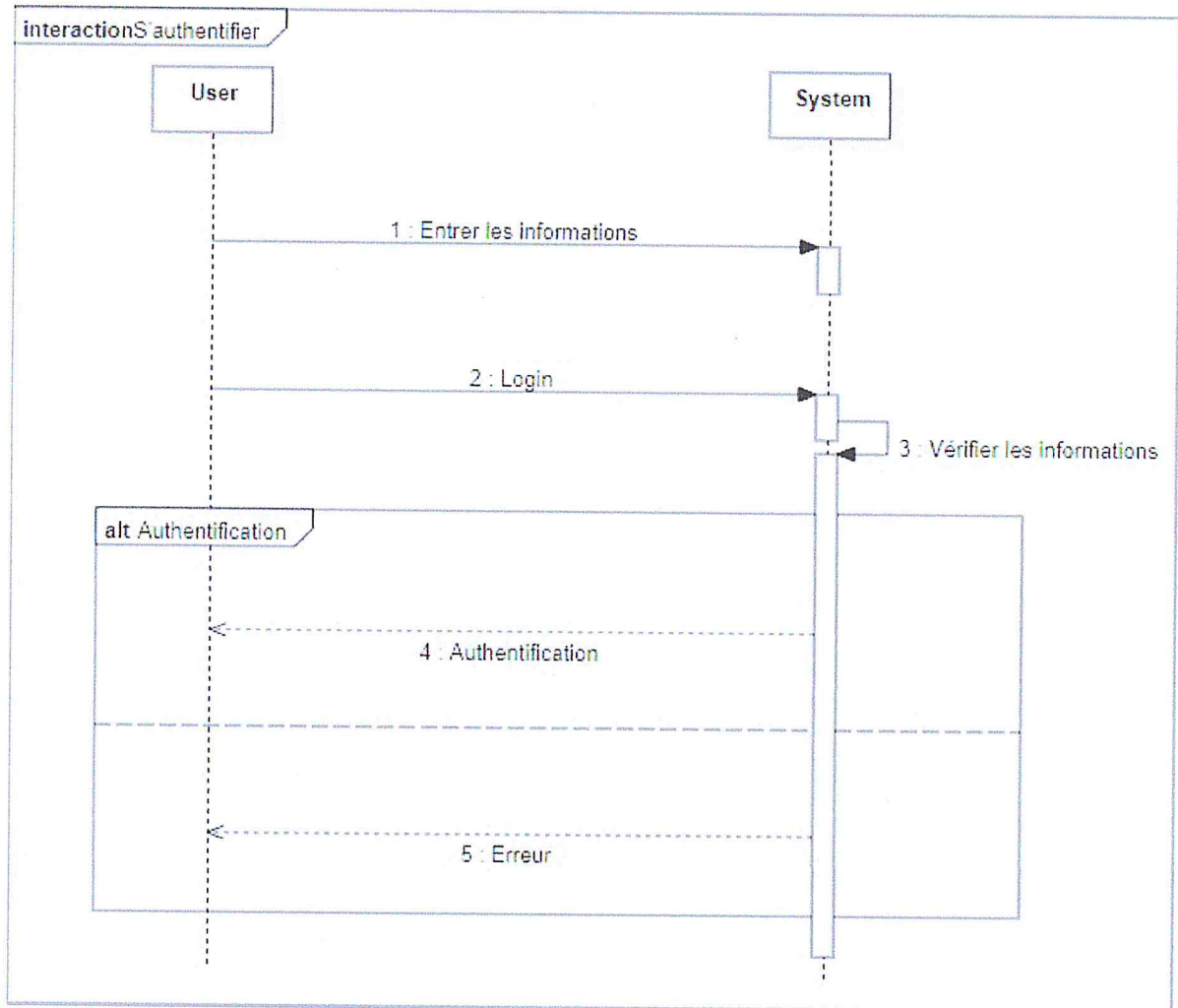


FIGURE 2.7: Diagramme de séquence de la cas d'utilisation "S'authentifier"

B. Ouvrir une synoptique

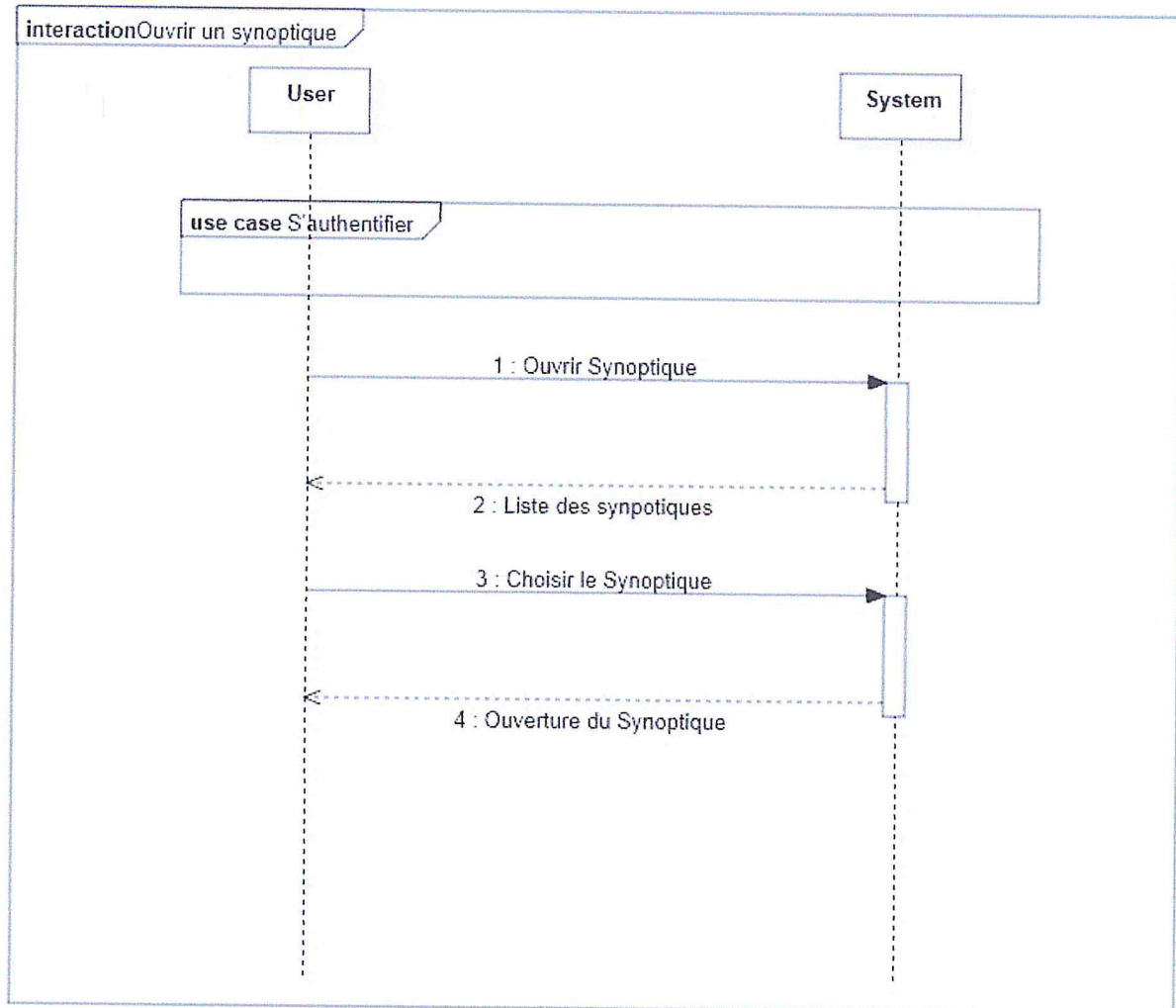


FIGURE 2.8: Diagramme de séquence de la cas d'utilisation "Ouvrir une synoptique"

C. Choisir une synoptique

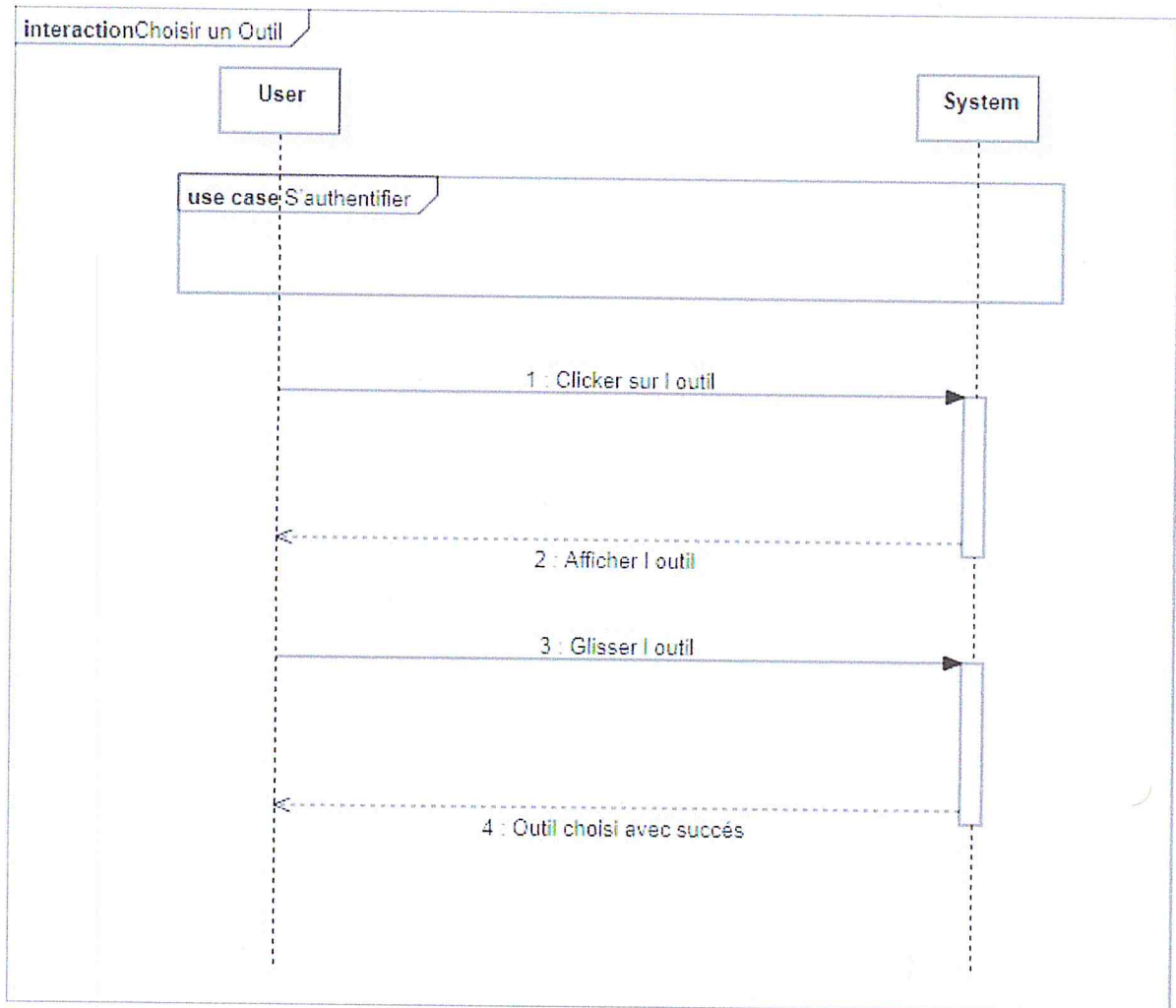


FIGURE 2.9: Diagramme de séquence de la cas d'utilisation "Choisir une synoptique"



CHAPITRE

— III —

RÉALISATION DE L'ÉDITEUR
GRAPHIQUE 3D

I. La modélisation 3D

I.1. Définition

Parlons d'abord sur la 3D, dans un premier temps il faut distinguer les types des dimensions utilisés : La dimension géométrique et la dimension mathématique. Dans cette partie nous sommes entrain de parler sur les dimensions géométrique.

La dimension géométrique correspond aux attributs géométrique théorique de base des objets utilisés qui sont :

1. Le point : Pas de dimensions (0D).
2. La ligne : Une seule dimension qu'elle est la longueur (1D).
3. La surface : Deux dimensions dont la longueur et la largeur (2D).
4. Le volume : Trois dimensions dont la longueur, la largeur et la hauteur - se dite aussi l'épaisseur ou bien la profondeur - (3D).

La 3D se fait sur un écran. C'est une image qui possède une longueur, une largeur et une profondeur. La 2D est la représentation selon deux axes, elle donne un effet d'aplatissement, alors que la 3D est une représentation selon 3 axes, elle donne un effet très réaliste, proche de l'espace dans lequel nous vivons.

Grâce à la perspective, elle permet de représenter des objets en trois dimensions sur une surface plane (2 dimensions), en tenant compte des effets d'éloignement et de leurs positions dans l'espace par rapport à l'observateur.[12]

Donc c'est quoi la modélisation 3D?, D'après qu'est-ce qu'on a vu dans le premier chapitre (le chapitre Introduction), la modélisation 3D est une technique infographique qui fait partie du domaine informatique, et qui permet de créer des objets en trois (03) dimensions dans un ordinateur, et permet aussi de réaliser des objets complexes, des environnements, des personnages, des bâtiments, des paysages, etc. à partir de formes simples telles que des sphères, des polygones, des cônes, des cylindres, des pyramiques, des cubes ou encore des courbes qui définissent les structures.

Le modélisation 3D peut être utilisée dans l'industrie automobile pour créer le design de nouveaux modèles, et aussi dans l'architecture pour établir des plans et se rendre d'avantage compte des espaces intérieurs et extérieurs, elle est aussi peut être utilisée dans les sciences pour présenter des phénomènes médicaux ou bien pour

recréer le corps humain de manière virtuelle, ainsi dans ce qui concerne les jeux vidéo et les films d'animations pour créer des personnages et des décors.

I.2. Les avantages

La modélisation 3D a quatre (04) principaux avantages qui sont :

1. L'universalité de l'outil.
2. Les capacités d'analyse, d'aide à la compréhension et à l'interprétation.
3. Un outil d'aide à la décision et de suivi des travaux.
4. L'aspect didactique : Il se situe à plusieurs niveaux, l'enseignant peut l'utiliser dans ses cours afin d'illustrer les notions structurales qu'il tente d'inculquer, elle est aussi plus parlante que toutes les explications orales.

I.3. Les inconvénients

Comme la modélisation 3D a des avantages, elle a aussi quelques inconvénients qui sont les suivants :

1. Le temps consacré à l'introduction des données.
2. Le manque de connaissance de quelques cas étudiés.

I.4. Les outils de la modélisation 3D

Les modèles 3D sont des composants essentiels du processus de production numérique. C'est pourquoi choisir le bon logiciel 3D est important, ce que nous permettra de mener à bien nos projets et donner vie à nos idées sans trop d'effort.

Il existe plusieurs logiciels qui s'intéressent à la modélisation trois dimensionnelle, quelques uns sont destinés aux débutants, autres aux intermédiaires, autres aux professionnels et autres aux industriels. Quelques uns sont aussi gratuits et les autres sont des logiciels payants.

Dans ce paragraphe, on va citer quelques logiciels qui sont les plus importants dans la modélisation 3D :

- **TinkerCAD** : Destiné au débutant, c'est une application en ligne pour modéliser en 3D, Il permet de modéliser des objets par glisser-déposer de formes géométriques de base, qui peuvent ensuite être modifiées, fusionnées ou sous-traitées pour créer des objets complexes 3D.
- **SketchUP** : Ce qu'on a utilisé dans notre projet, il est facile à utiliser destiné au débutants, Il prend également en charge des extensions permettant une modélisation plus rapide ou des fonctionnalités supplémentaires pour des champs d'application spécifiques.
- **Blender** : Il est aussi utilisé dans notre projet, il est devenu absolument incontournable et d'excellente qualité, Blender est une suite gratuite et open-source de modélisation 3D extrêmement complète, il dispose par ailleurs d'une bibliothèque très fournie d'extensions qui permettent d'ajouter des générateurs (d'arbres, de nuages, de particules, de paysages...), des options de modélisation, des objets, des scripts, des options d'importation ou d'exportation (depuis ou vers AfterEffects, Unreal Engine, DirectX, etc.).
- **123D Sculpt+** : Est une application de modélisation 3D sur les tablettes tactiles (Android et iOS), Édité par Autodesk. L'utilisateur choisit quelques formes basiques pour commencer, qu'il peut ensuite les modifier et les assembler pour former des créations plus complexes. Il peut aussi importer des objets depuis une bibliothèque fournie, et sauvegarder ses créations sur le Cloud.
- **Maya et 3D Studio Max -3DS Max-** : De niveau professionnel, Maya est détenu par Autodesk, qui détient aussi son concurrent 3DS Max, moins cher et moins complet, mais aussi plus répandu dans l'industrie du jeu vidéo.
- **Cinéma 4D** : Comme son nom l'indique, Cinema 4D est avant tout un logiciel de création 3D fait pour la vidéo ultra réaliste destinée à la télévision ou au cinéma, même s'il est aussi parfois utilisé pour les jeux vidéo. Plus fluide et simple à prendre en main par rapport à 3DS Max.
- **SolidWorks** : Édité par Dassault Systems, SolidWorks est la solution de conception 3D assistée par ordinateur (CAO 3D) la plus populaire du moment. Le

logiciel industriel permet de concevoir des pièces détaillées, des assemblages et des mises en plan destinées à la production, et dispose d'outils permettant la génération de surfaces complexes, de pièces de tôlerie dépliées et d'assemblages mécano-soudés. Solidworks permet en outre de réaliser des analyses de contraintes ou encore d'évaluer l'impact environnemental d'un objet en fonction de ses composants.[13]

I.5. Les logiciels utilisés dans notre projet

Comme on a vu, pour faire la modélisation tridimensionnelle il faut utiliser des logiciels (ou application) spécialistes dans la modélisation 3D. Dans notre travail - comme on vous a montré précédemment - on a utilisé comme logiciel de modélisation 3D **SketchUP** et **Blender** parcequ'ils sont faciles à utiliser et aussi ils ont des avantages par rapport aux autres logiciels de modélisation 3D qu'on va les voir juste après.

A. SketchUp

Destiné aux phases conceptuelles du dessin, SketchUp est un logiciel 3D à la fois puissant et facile à apprendre. Il est considéré un peu comme le crayon du dessin numérique. Ce logiciel maintes fois primé rassemble un ensemble d'outils à la fois simples et complets qui permettent de rationaliser et de simplifier le dessin en 3D sur votre ordinateur. SketchUp est utilisé par tous ceux qui souhaitent rêver, concevoir et communiquer en 3D.

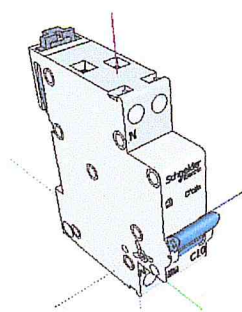


FIGURE 3.1: Exemple de disjoncteur édité par SketchUp.

A.a. Historique

A ce moment là, SketchUp fait parti de Google, mais il n'est pas été créer à l'origine de Google, car en fait il a été présenté pour la première fois par *Last Software*¹, avec une date de sortie originale en août 2000, Schell et son équipe ont développé SketchUp comme outil de modélisation 3D pour les architectes, les concepteurs et les cinéastes. Comme Digital Media Net a décrit : "SketchUp de *Last Software* a été motivé par le besoin croissant des professionnels de la conception d'un programme de modélisation 3D plus intuitif et accessible. Le cœur de SketchUp est une interface dans laquelle l'utilisateur dessine simplement les bords du modèle souhaité dans l'espace 3D et le logiciel «remplit» automatiquement les formes pour créer une géométrie 3D".[14]

Le mars 2006 et après le lancement réussi de Sketchup, *Last* a voulu permettre aux gens de pouvoir placer les modèles qu'ils ont créés dans Sketchup dans Google Earth. Ils ont collaboré avec Google pour développer un plug-in pour Google Earth, afin de permettre aux gens de «géolocaliser» leurs modèles dans Google Earth.

En janvier 2007, Google a lancé SketchUp v.6 en faisant de version, l'une est payante pour 495 \$ et l'autre est gratuite. Et à la fin de la même année, la version 8 de SketchUp qui comprend de nombreuses améliorations et de nouvelles fonctionnalités est sorti.

Depuis juin 2012, SketchUp a été racheté par *Trimble Navigation*. Suite à cet rachat, le logiciel est mis à jour chaque année.

A.b. Avantages

Très à l'aise dans les phases de conception d'un projet, SketchUp présente de nombreux attraits pour ses utilisateurs qui sont les suivants :

- Il est très facile à utiliser : C'est l'avantage le plus gros de SketchUp par rapport aux autres logiciels de modélisation tridimensionnelle.
- Adapté à de nombreux usages et à de nombreux domaines d'activité comme l'architecture, la décoration, la menuiserie, l'organisation d'événements, etc.
- Sketchup est un logiciel extrêmement populaire dans le monde de la 3D et du Dessin Assisté par Ordinateur - DAO -.

1. Une société de technologie co-fondée en 1999 par Brad Schell.

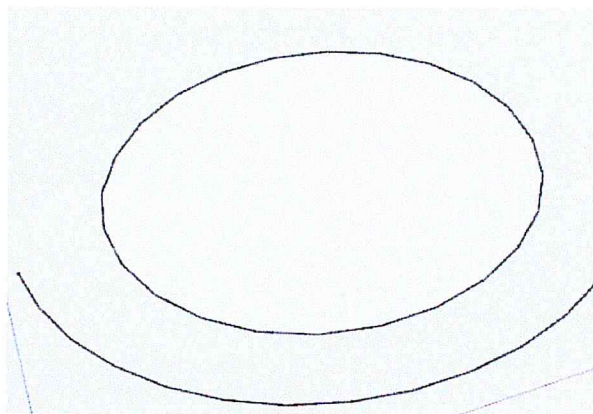


FIGURE 3.3: Exemple d'arc et cercle en SketchUp.

A.d. Comment on a utilisé SketchUp?

L'avantage principal qui nous a poussé à utiliser **SketchUp** ce qu'il est vraiment très très facile à utiliser et contient une grande bibliothèque gratuite qui nous fournit les différents types d'équipements 3D appelée **3D Warehouse**.

Cette bibliothèque était le vrai fournisseur des outils de notre projet, le travail fait dans cette partie est qu'on télécharge l'équipement en format ".Col" (COLLADA) ainsi ".Obj" et l'importe dans notre code source JavaFx.

B. Blender

Blender est un logiciel professionnel de modélisation 3D gratuit et open source grâce auquel vous pouvez créer des films d'animation, des effets visuels, des applications interactives, des jeux vidéo et aussi des modèles 3D.



B.a. Historique

Le début de Blender c'était à l'année 1995 par le studio d'animation néerlandais NeoGeo.

L'année 1998 était l'année de fondation d'une nouvelle société dénommée *Not a Number* (NaN), en tant que filiale de NeoGeo dédiée au développement et à la commercialisation de Blender. Son objectif premier était de créer et distribuer gratuitement une suite de création graphique 3D compacte et multiplateforme. C'était à l'époque

un concept révolutionnaire, car la plupart des logiciels commerciaux de modélisation coûtaient des milliers de dollars. *NaN* souhaitait mettre des outils de modélisation et d'animation 3D de niveau professionnel à disposition du grand public. Le modèle économique de *NaN* consistait à fournir produits et services commerciaux autour de Blender.

À l'été 2000, Blender 2.0 fut lancé. Cette version de Blender ajouta l'intégration d'un moteur de jeu à l'application 3D. Vers la fin de 2000, le nombre d'utilisateurs enregistrés sur le site web de *NaN* dépassa les 250 000.

La fin de l'année 2001, le premier logiciel commercial de *NaN*, *Blender Publisher*, était lancé. Ce produit visait le marché émergent des médias 3D interactifs sur Internet. Le dimanche 13 octobre 2002, Blender est remis à la communauté sous les termes de la GPL (GNU General Public License). Le développement de Blender se poursuit depuis lors, conduit par une équipe de bénévoles dévoués du monde entier, sous la houlette de son créateur, *Ton Roosendaal*. [15]

B.b. Avantages

Le logiciel Blender a plusieurs avantages qui sont les suivants :

- Blender est un logiciel libre et open source (logiciel libre sous licence GPL) : Grâce auquel on peut créer des films d'animation, des effets visuels, des applications interactives, des jeux vidéo et bien sûr, des modèles 3D.
- Il fournit un certain nombre d'outils et de modificateurs qui facilitent la création de maillages destinés à l'impression 3D, avec notamment la possibilité de réparer ces derniers : A cause de ça, on peut monter, composer, sonoriser des séquences complexes mêlant vidéo 3D, 2D, son et mêmes prises de notes, sur une même interface commune et adaptable.
- Il a une architecture 3D de haute qualité, permettant un flux de travail de création rapide et efficace.
- Il a un exécutable de petite taille, facultativement portable.
- La possibilité d'exporter des fichiers Blender vers d'autres formats.

B.c. Inconvénients

BLENDER reste en effet pour le moment une solution, certes attrayante, mais qui n'arrive pas à s'imposer dans les écoles et les studios d'animation professionnels.

B.d. Comment on a utilisé Blender ?

Blender est un logiciel professionnel libre et open source qui a une architecture de haute qualité et qui presque n'a pas de points faibles.

Dans notre projet, Blender était un élément efficace dans le découpage des équipements en prenant ceux que nous avons besoin de ces eux. Ce logiciel nous donne la possibilité d'identifier le morceau coupé pour qu'on puisse le connaître au cour de la programmation.

Comme c'est exprimé ci-dessous dans les figures 3.4 et 3.5 on a retiré le composant cadré en bleu en le donnant un identifiant pour l'appeler dans la partie programmation par cet identifiant.

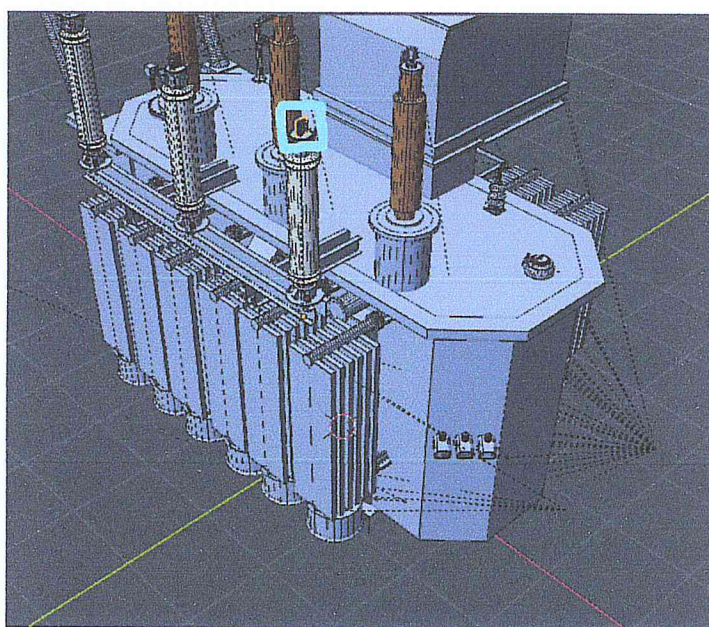


FIGURE 3.4: Figure exprime la structure compète d'un composant et la partie retirée.

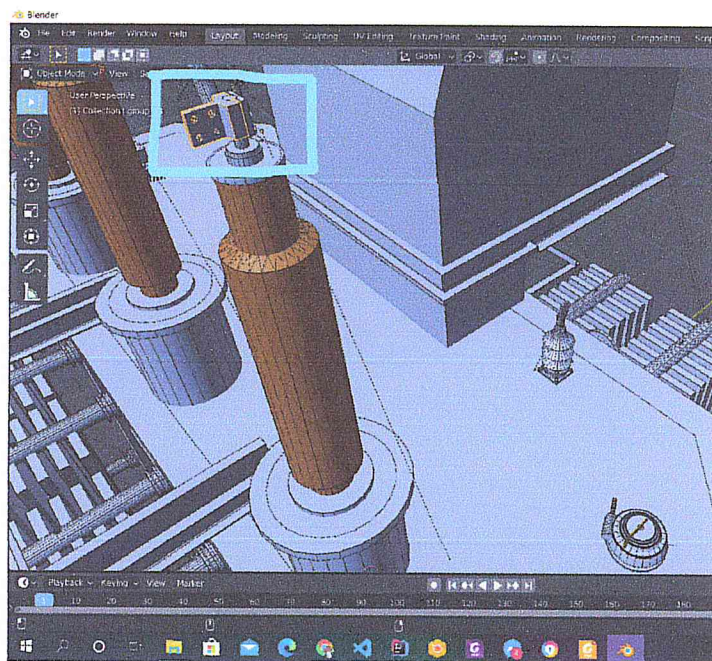


FIGURE 3.5: Figure exprime le découpage du composant précédent en le donnant un identifiant.

I.6. L'architecture MVC

Cette architecture est le premier essai de séparation de l'interface utilisateur. Le pattern MVC - *Model View Controller* - permet de bien organiser le code source. Il aide à savoir quels fichiers créer, mais surtout à définir leur rôle. Le but de MVC est justement de séparer la logique du code en trois parties que l'on retrouve dans des fichiers distincts :

- Le modèle.
- La vue.
- Le contrôleur.

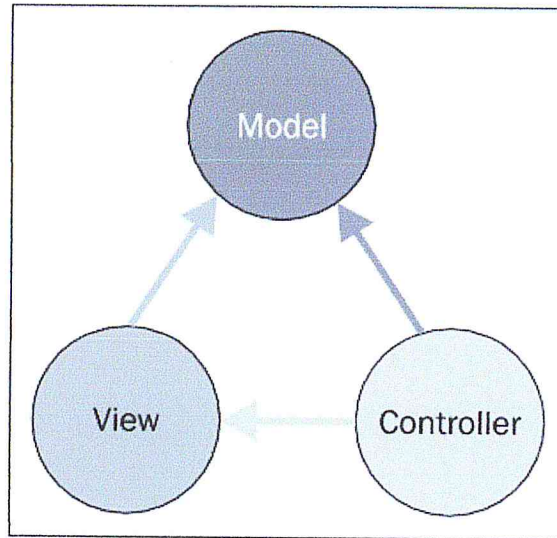


FIGURE 3.6: L'architecture MVC.

.a. Le modèle

Il représente le fond (le sujet d'étude).

Son rôle est d'aller récupérer les informations à partir de la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur, donc il contient les données manipulées par le programme.

.b. La vue

Cette partie concerne principalement la représentation des données du modèle à un écran (ou sur tout autre périphérique de sortie), à cause de ça qu'on peut avoir plus d'une seule vue.

Elle a comme rôle de représenter la donnée encapsulée via un modèle et de se maintenir à jour lorsque le modèle est modifié, par conséquent elle doit s'enregistrer comme écouteur au niveau du modèle, donc elle interagit avec le modèle.

.c. Le contrôleur

C'est lui qui va demander au modèle les données, les analyser, prendre des décisions et renvoyer le texte à afficher à la vue, donc il permet à l'utilisateur de modifier la donnée encapsulée dans le modèle.

A. Comment ces éléments communiquent entre eux?

Il faut tout d'abord retenir que le contrôleur est le chef d'orchestre : c'est lui qui reçoit la requête du visiteur et qui contacte d'autres fichiers (le modèle et la vue) pour échanger des informations avec eux.

Le fichier du contrôleur demande les données au modèle sans se soucier de la façon dont celui-ci va les récupérer. Par exemple : « Donne-moi la liste des 30 derniers messages du forum no 5 ». Le modèle traduit cette demande en une requête SQL, récupère les informations et les renvoie au contrôleur.

Une fois les données récupérées, le contrôleur les transmet à la vue qui se chargera d'afficher la liste des messages.

Dans les cas les plus simples, le contrôleur sert seulement à faire la jonction entre le modèle et la vue. Mais le rôle du contrôleur ne se limite pas à cela : s'il y a des calculs ou des vérifications d'autorisations à faire, des images à miniaturiser, c'est lui qui s'en chargera.

Concrètement, le visiteur demandera la page au contrôleur et c'est la vue qui lui sera retournée, comme schématisé sur la figure suivante. Bien entendu, tout cela est transparent pour lui, il ne voit pas tout ce qui se passe sur le serveur.[16]

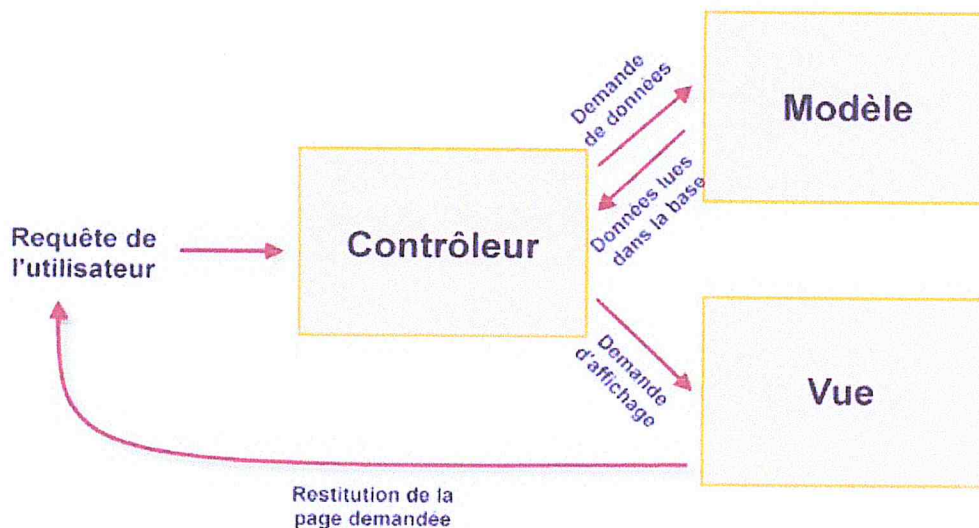


FIGURE 3.7: La requête du client arrive au contrôleur et celui-ci lui retourne la vue.

B. Avantages

- Possibilité de mettre plusieurs vues pour un seul modèle.
- La facilité d'ajouter une vue.
- Un gain de temps de maintenance et d'évolution.
- Diminution de la complexité lors de la conception.[17]

C. Inconvénients

- L'inconvénient majeur du modèle MVC n'est visible que dans la réalisation de petits projets, parce que dans les petits projets on est pas besoin de faire toute cette architecture qui prend beaucoup du temps et qui n'a aucun effet dans ce cas là.

CHAPITRE

— IV —

DESCRIPTION DE L'ÉDITEUR RÉALISÉ

I. L'interface de l'éditeur

I.1. Introduction

L'éditeur graphique 3D pour le système de contrôle, de supervision et d'acquisition des données SCADA était programmé en JavaFx et basé sur la base de données de SketchUp *3D Warehouse* qui a fourni à notre système les composants électriques nécessaires - qui sont cités juste après - sachant que chaque composant a la possibilité de se connecter avec un autre composant en reliant entre eux avec une ligne électrique 3D :

- Disjoncteur.
- Générateur.
- Transformateur de courant.
- Transformateur 3 enroulements.
- LED.
- Masse.
- Porte.
- Parafoudre.
- Compartiment de contrôle.
- Pylône.

I.2. Le contenu de l'interface graphique de l'éditeur

Comme nous montre la figure ci-dessous l'interface graphique de notre éditeur contient - Comme c'est numéroté dans la figure - :

1. Un petit panneau qui retourne la structure du composant sélectionné de la liste en 3D.
2. La liste des composants.
3. Le panneau où il doit être affiché le plan 3D qui contient les composants sélectionnés de la liste.
4. Les flèches dirigeant de l'angle de vue.

5. Un espace pour passer au système les informations nécessaires du composant sélectionné.
6. La barre des boutons rapides.

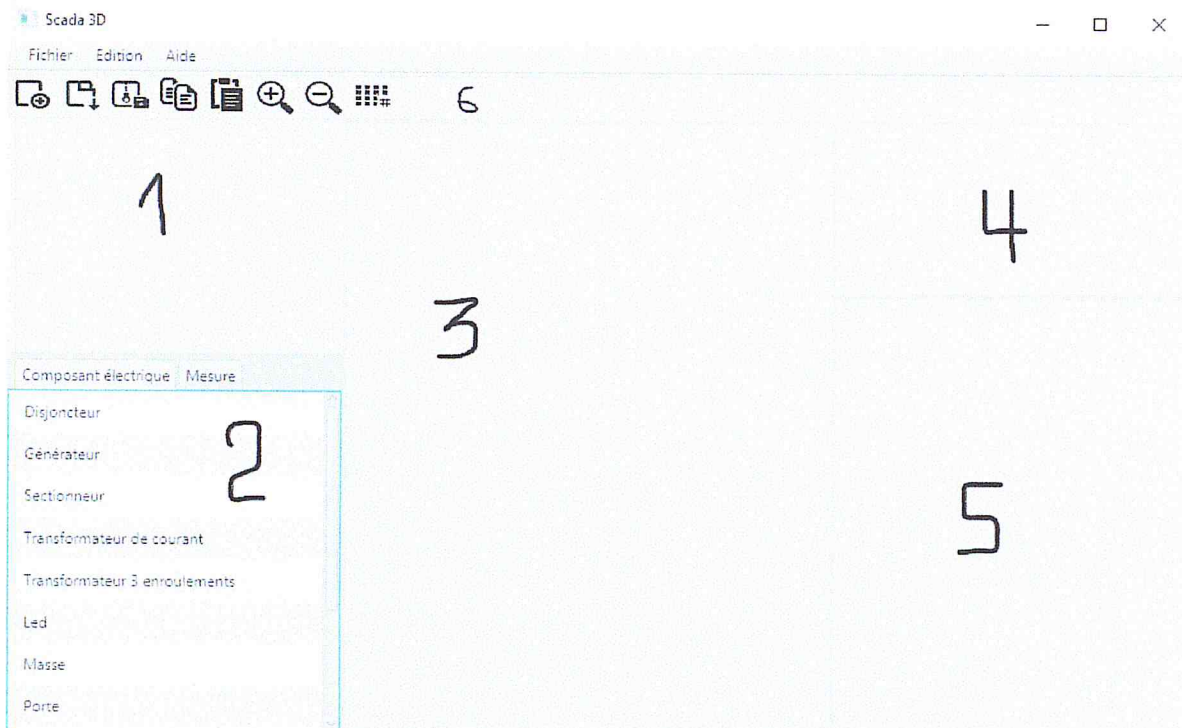


FIGURE 4.1: Une capture d'écran de l'interface graphique de l'éditeur 3D.

I.3. Description de l'interface

A. Création du schéma

Pour que l'utilisateur peut créer son propre schéma graphique 3D, il doit d'abord créer une synoptique en cliquant sur le premier bouton à gauche de la barre des boutons rapides ou bien en cliquant sur "Fichier" dans la barre d'outils et après sur "nouveau".

Après la création de synoptique, un espace avec un plan 3D - où il se pose les composants sélectionnés par l'utilisateur - sera créer, comme c'est affiché dans la figure 4.2 ci-dessous.

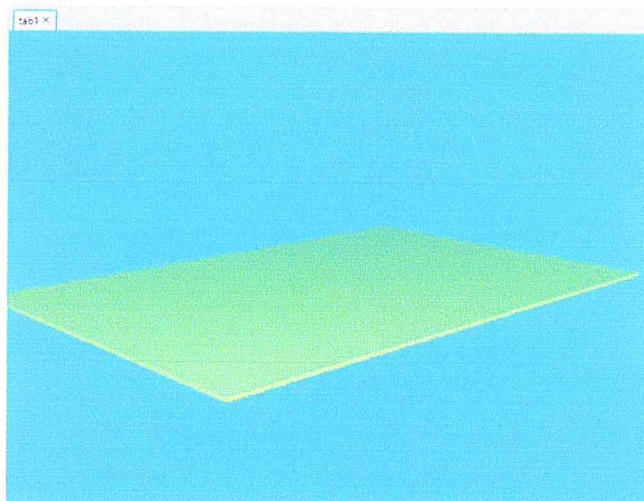


FIGURE 4.2: Une capture d'écran de l'espace où s'affiche notre schéma 3D.

B. Les informations des composants

Dès que le plan 3D est affiché, l'utilisateur doit avoir la permission de sélectionner les composants qu'il souhaite les afficher sur son schéma à partir de la liste des composants électriques sachant que dès que l'utilisateur se clique sur un de ces composants une image 3D doit afficher la structure du composant sélectionné comme c'est exprimé dans la figure suivante.

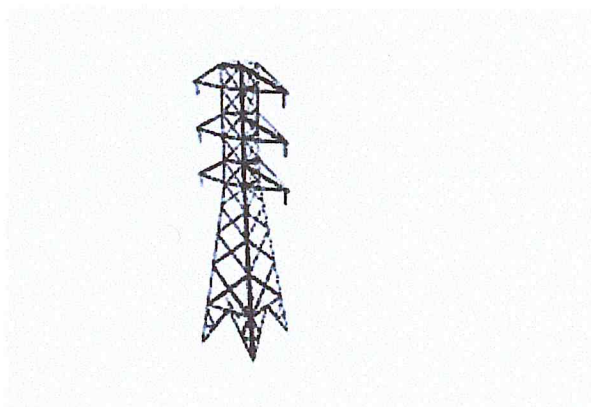


FIGURE 4.3: Une capture d'écran de l'espace où s'affiche la structure du composant sélectionné.

Le composant électrique sélectionné qui doit être affiché dans le schéma graphique 3D a un ensemble d'information que l'utilisateur doit saisir dans le panneau affiché à droite du schéma graphique, ce qu'il est exprimé dans la figure 4.5, sachant que dans le haut de ce panneau il s'affiche le nom de l'objet sélectionné (Dans la figure c'était un générateur).

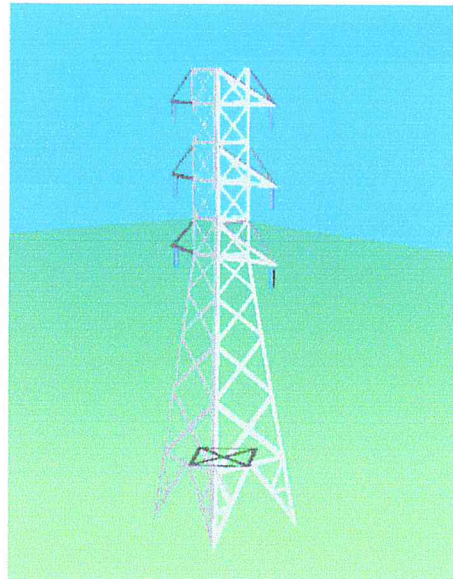
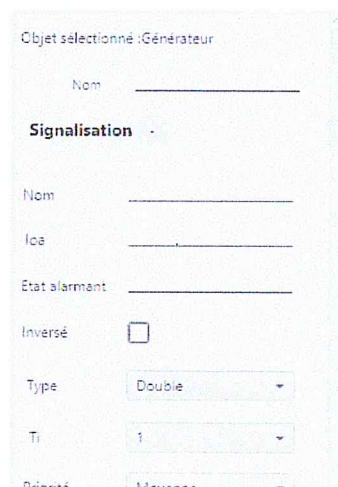


FIGURE 4.4: Une capture d'écran d'un exemple du composant 3D (Pylône).



Objet sélectionné : Générateur

Nom _____

Signalisation

Nom _____

Ice _____

Etat alarmant _____

Inversé

Type Double ▼

Ti 1 ▼

Préité Moyenne ▼

FIGURE 4.5: Une capture d'écran de l'espace où l'utilisateur doit saisir les informations de chaque composant.

La saisie des informations du composant impose l'enregistrement automatique de ces informations, c'est-à-dire que l'utilisateur n'est pas besoin de cliquer sur un bouton

sauvegarder pour qu'il puisse sauvegarder ces informations, mais dès qu'il sauvegarde le schéma 3D qu'il a créé elles devront être enregistrées automatiquement.

Important : En raison des conditions que le monde traverse depuis le déclenchement de l'épidémie de Covid-19 et en raison de la quarantaine imposée par les autorités locales, qui a provoqué la fermeture de nombreuses entreprises et usines, dont Elit, nous n'avons pas pu obtenir de nombreuses informations relatives aux composants électriques de l'entreprise. Par conséquent, et jusqu'au moment de la rédaction de ces lignes là, cette étape n'est pas encore faite, sachant qu'elle le sera dès la levée de la quarantaine et que l'entreprise reprendra son travail normalement.

Cette éditeur graphique 3D donne aussi la possibilité à l'utilisateur de manipuler sur le schéma qu'il a déjà créé à partir les flèches dirigeantes qui sont aperçues dans la figure suivante.

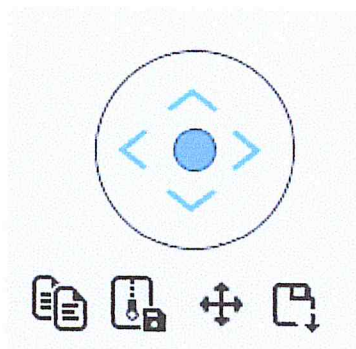


FIGURE 4.6: Une capture d'écran qui montre les flèches dirigeantes.

C. La connectivité des composants

Après que l'utilisateur insère un ensemble de composants électriques 3D, il devra besoin de connecter entre eux en utilisant des cables électriques et ils devront être toujours connectés meme si l'utilisateur a changé l'emplacement de chacun d'uns.

La figure suivante nous montre un exemple de deux (02) composants électriques de la bibliothèque 3D qui sont connectés entre eux avec un cable électrique.



FIGURE 4.7: Une capture d'écran qui montre la connectivité entre deux (02) composants.

D. La sauvegarde des schémas 3D

L'utilisateur après qu'il a créé le schéma 3D qu'il veut, il est besoin de sauvegarder ce qu'il a fait pour qu'il puisse l'utiliser autre fois, cette opération (Sauvegarde) existe dans notre éditeur graphique 3D il suffit juste que l'utilisateur faire un clique de souris sur le bouton "Sauvegarder" dans la barre des boutons rapides ou bien par un clique sur "Fichier" et après sur "Sauvegarder" dans la barre d'outils, par ça le schéma doit être enregistré en extension '.SVG'.

Remarque :

SVG signifie *Scalable Vector Graphics*, en clair c'est un format de données utilisé pour définir des graphiques vectoriels.

Ce format est basé sur du XML et permet de définir des éléments graphiques pour le web. Il est surtout utilisé pour l'affichage de graphiques.

Le principal avantage de ce format est que l'on peut redimensionner une image sans aucune perte de qualité contrairement à une image jpeg ou png par exemple. Ceci est dû au fait que ces images soient vectorielles.[18]

E. Comparaison entre une modélisation 2D et une modélisation avec l'éditeur graphique 3D

La figure ci-dessous exprime un petit exemple de schéma 2D faite en utilisant l'éditeur 2D de Elit.

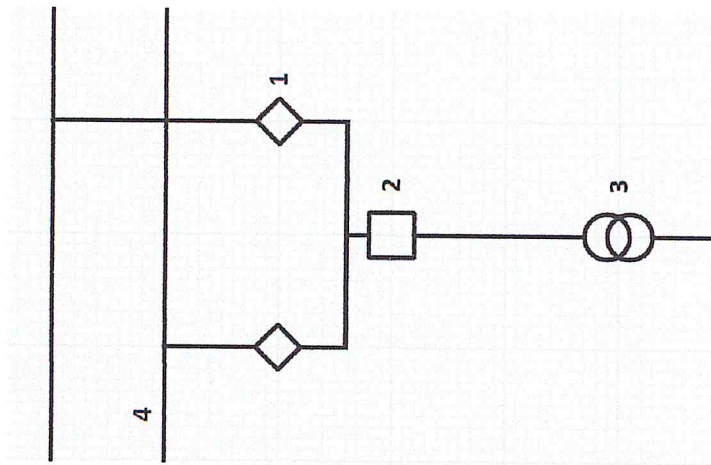


FIGURE 4.8: Un schéma 2D faite par l'ancien éditeur 2D de Elit.

Ce schéma se compose par les éléments suivants :

1. Sécionneur.
2. Disjoncteur.
3. Transformateur.
4. Les jeux de barre.



FIGURE 4.9: Le schéma 2D précédent faite par notre éditeur 3D.

Les deux (02) figures ci-dessus nous montre clairement la différence entre la modélisation 2D et l'autre 3D.

Comme on voit dans les deux (02) figures, la modélisation avec l'éditeur 3D est plus réelle que l'autre en 2D, et laisse l'utilisateur pense qu'il est vraiment dans le terrain de travail.

F. Les caractéristiques de notre éditeur graphique 3D

L'éditeur graphique 3D pour le système de contrôle, de supervision et d'acquisition des données SCADA est un éditeur qui modélise des schémas et des plans d'électricité d'Elit en 3D sachant que l'existant maintenant à Elit est un éditeur en 2D.

Notre éditeur est dédié spécialement à Elit, donc il répond juste au besoin demandé par Elit, ça veut dire que le nombre de besoin à modéliser est limité. par conséquent, on voit que c'est impossible à faire ce travail dans un autre éditeur graphique 3D qui existe déjà et aussi parce que ce nombre de besoins à modéliser (La bibliothèque intégrée dans l'éditeur) doit avoir un ensemble d'informations affectées spécialement à un seul composant, ce qu'il est impossible de le faire dans un autre éditeur.

CONCLUSION

Le groupe Sonelgaz dans ce moment là utilise un système de contrôle, de supervision et d'aquisition des données en temps réel SCADA pour le pilotage et le suivi de son réseau électrique. Ce système est représenté par un éditeur graphique 2D qui consiste à créer des schémas avec des composants prédéfinis en 2D.

comme on sait bien que la modélisation 2D et malgré qu'elle n'est pas cher, mais elle ne ressemble pas beaucoup à la réalité. A cause de ça, notre travail consiste à créer un éditeur graphique 3D qui consiste à faire le même travail que l'autre en 2D mais juste en 3D pour rendre la vision plus réelle.

La réalisation de cette éditeur a passé par les étapes que chaque projet passe par les. Notre éditeur consiste à faire des diagrammes et des schémas synoptiques en 3D avec des composants et des symboles prédéfinis.

Notre éditeur répond presque au besoins de Elit (les besoins non faits sont des besoins relies au système de l'entreprise tels que l'affectation des points à partir de la fiche TélÉInfo et les affectations des informations au composants) et il a la possibilité d'être amélioré mais il sera bien que notre équipe soit assistée par un architecte designer qui maitrise bien les logiciels de la modélisation 3D tels que Blender et SketchUp.

BIBLIOGRAPHIE

- [1] D. M. N. Jeffrey J. Farris, "Evaluation of secure peer-to-peer overlay routing for survivable scada systems," *Department of Electrical and Computer Engineering University of Illinois, Urbana-Champaign Urbana, Illinois 61801, U.S.A.*, 2004.
- [2] M. Electronics, "all about circuits." <https://www.allaboutcircuits.com>.
- [3] J. W. B. M. R. G. Robert D. Larkin, Juan Lopez Jr, "Evaluation of security solutions in the scada environment," *The DATA BASE for Advances in Information Systems*, p. 2, November 2014.
- [4] T.-h. K. Rosslin John Robles, *Application Program Interface as Back-Up Data Source for SCADA Systems*. Multimedia Engineering Department, Hannam University, Daejeon, Korea.
- [5] P. RAYMOND, "Les automates programmables industriels (api)." Octobre 2005.
- [6] H. N. Akira Sugimoto, Tatsuro Sano, "Proposal of generating and customizing method to construct 3d gui for scada," *Théorie de l'électricité C*, vol. 128, 2008.
- [7] Sonelgaz, "Présentation du groupe sonelgaz," 2020. <https://www.sonelgaz.dz/fr/category/qui-sommes-nous>.
- [8] Elit, "Qui sommes nous?," 2020. <https://www.elit.dz/639/qui-sommes-nous>.

-
- [9] D. M. E. B. ABBAS Kamel, "Etude, modélisation et construction d'un prototype de smart cities avec esri cityengine," *UNIVERSITE ABDELHAMID IBN BADIS - MOSTAGANEM, Faculté des Sciences Exactes et de l'Informatique, Département de Mathématiques et d'Informatique, Filière : Informatique*, 2016.
- [10] L. Audibert, *UML 2 - De l'apprentissage à la pratique*.
- [11] UML, "Présentation des diagrammes uml 2.5." <https://www.uml-diagrams.org/uml-25-diagrams.html>.
- [12] A. C. Muro, "La 3d dans le monde cinéma." <http://troisdimension.e-monsite.com/pages/introduction/introduction.html>.
- [13] G. Champeau, "Modélisation 3d : les meilleurs logiciels et applications," 30 Novembre 2015. <https://www.numerama.com/tech/132900-logiciels-et-outils-de-modelisation-3d.html>.
- [14] M. SketchUp, "History of sketchup." https://masterSketchUp.com/History_of_SketchUp.
- [15] Blender, *La documentation de Blender*. Blender.Doc.fr.
- [16] C. de terminale SI, *Architecture MVC*. 5-architecture-MVC.odt.
- [17] J.-M. Richer, *Developpement Web Le modele MVC*. Université Angers, faculté des séances, unité de formation et de recherche, 24 Janvier 2011.
- [18] SYNBIOS, "Le svg, pour quoi faire?." <https://www.synbioz.com/blog/tech/le-svg-pour-quoi-faire>.