

RAPPORT DU PROJET DE FIN D'ÉTUDES

MASTER INFORMATIQUE, FONDEMENT ET INGÉNIERIE

Parcours

WEB

Security Requirements Engineering Platform

Melle Fadoua BERREHILI
Mr Mohammed BOURAGBA

Sous la direction de :
M. Yves ROUDIER

Membres de jury :
M. Michel RIVEILL
M. Molines GUILHEM

Le 26 janvier 2018

Security Requirements Engineering Platform

Soutenu par :

M.lle Fadoua BERREHILI

Mer Mohammed BOURAGBA

Membres de jury :

M. Michel RIVEILL

M. Molines GUILHEM

Dédicaces

Ce travail, fruit de nos efforts tout au long de la période du PFE, est dédié aux personnes qui nous ont toujours soutenue :

A nos Parents,

Qui nous ont tant donné sans jamais se lasser, ceci n'est qu'une infime partie de tout ce que nous avons reçu d'eux. On les aime infiniment.

A nos frères et sœurs ,

Source de notre fierté, une famille plus que formidable, on prit le bon Dieu pour qu'il préserve notre forte union et qu'il renforce notre grand amour.

A nos ami(e)s,

Source de motivation inépuisable, que serait notre vie sans vous ? on ne peut même pas l'imaginer !! Que notre amitié perdure pour toujours.

Remerciements

Il nous est agréable de nous acquitter d'une dette de reconnaissance auprès de toutes les personnes, dont l'intervention au cours de ce projet, a favorisé son aboutissement.

On tient à exprimer notre gratitude à M. Yves ROUDIER , notre encadrant, pour ses directives précieuses et ses conseils pertinents qui nous ont appuyé considérablement pendant toute la période du projet.

Que messieurs les membres de jury trouvent ici l'expression de nos reconnaissances d'avoir accepté de juger notre travail. On profite de cette occasion pour remercier l'ensemble du corps professoral du master Informatique, fondement et ingénierie à l'école polytech Nice sophia antipolis pour leurs efforts en vue d'assurer une formation de haut niveau pour l'ensemble des étudiants.

Que tous ceux et celles qui ont contribué de près ou de loin à l'accomplissement de ce travail trouvent l'expression de nos remerciements les plus chaleureux.

Résumé

Selon l'AFIS (Association Française d'Ingénierie Système), l'ingénierie système peut se définir comme « Une démarche méthodologique interdisciplinaire qui englobe l'ensemble des activités adéquates pour concevoir, faire évoluer et vérifier un système apportant une solution économique et performante aux besoins d'un client tout en satisfaisant l'ensemble des parties prenantes ». Cette démarche intègre de nombreux processus standardisés par les normes IEEE 1220 ou ISO 15288 parmi lesquels nous retrouvons la définition et l'analyse des exigences, la définition des architectures fonctionnelle et structurelle et le processus de validation vérification. C'est dans ce but que les développeurs ont pensé à faciliter la phase de conception et de vérification aux concepteurs et cela en créant des applications qui permettent la réalisation de l'ensemble de ces démarches. Le présent rapport décrit l'étude et la réalisation d'une application qui englobe une partie de ces tâches.

Table des figures

1.1	Avantages SysML	4
1.2	Fonctionnalités de SysML	4
1.3	Les composants de SysML	5
1.4	Comparaison : UML et SysML	6
1.5	Vue d'en-tête d'un diagramme SysML	6
1.6	Caractéristique du diagramme paramétrique	7
1.7	Diagramme paramétrique	7
1.8	Diagramme d'exigence	8
1.9	Diagramme d'exigence, Machine à café	8
1.10	Etapes pour avoir une architecture valide	9
1.11	Définition TTool	10
2.1	La première interface de TTool	12
2.2	Choix du type de diagramme	13
2.3	Interface pour dessiner les diagrammes	13
2.4	Les composants du code source de TTool	14
3.1	Schéma du Modèle,Vue et Contrôleur [8]	18
4.1	Maquette de la page d'accueil de l'application	21
4.2	La maquette de l'éditeur des diagrammes	21
5.1	Première interface de TToolWEB	22
5.2	Onglet de l'éditeur TToolWEB	23
5.3	L'outil TToolWEB	23
5.4	Onglet Documentation	24
5.5	Onglet Contact	24
5.6	Composition de notre application	25

Table des matières

Dédicaces	i
Remerciements	ii
Résumé	iv
Table des figures	v
Introduction Générale	1
Chapitre 1 Contexte Général du Projet	2
1.1 Etat de l’art	2
1.2 Etude et familiarisation	3
1.3 Comparaison entre SysML et UML	5
1.4 SysML-sec	9
1.5 L’outil TTool	9
Chapitre 2 Etude du projet	11
2.1 Objectif	11
2.2 Problème rencontrés	12
Chapitre 3 Réalisation	15
3.1 Environnement de développement	15
3.1.1 Environnement matériel	15
3.1.2 Environnement logiciel	15
Chapitre 4 Prototypage	20
4.1 Maquettage	20
Chapitre 5 Démonstration	22
5.1 Notre application	22
5.2 Composition de notre application	24
Conclusion générale	27

Bibliographie	28
Table des matières	28

Introduction Générale

De nos jours, La complexité des nouveaux systèmes ne cesse de s'accroître, en termes d'intégration de multiples technologies, de nombre de composants, ainsi qu'en termes de performances attendues et en contraintes sécuritaires. De plus, les phases de conception de ces systèmes doivent également garantir la tenue de délais stricts pour un coût maîtrisé. D'autre part, le recueil des exigences joue un rôle déterminant pour les phases d'analyse, de conception, de réalisation et de maintenance qui lui font suite. La traçabilité des exigences au long du cycle de vie est ainsi devenue une préoccupation majeure des praticiens du domaine de l'ingénierie système. Ces mêmes praticiens reconnaissent l'importance d'une analyse d'exigences menée au plus tôt dans le cycle de vie d'un système.

Outre les relectures croisées qui permettent d'apprécier le degré de cohérence des exigences entre elles, il est un type d'analyse essentiel pour détecter au plus tôt les erreurs de conception et réduire le coût de développement et de test du système : la confrontation d'un modèle de conception aux exigences. En contexte SysML, ce type de confrontation met en balance une architecture d'instances de blocs et les comportements des entités qui la composent, avec les exigences exprimées dans les diagrammes d'exigences, les cas d'utilisation et paramétriques.

L'objectif de cette étude est de savoir comment utiliser la notation SysML (diagrammes d'exigence et paramétriques) dans le cadre d'un processus complet partant des premiers contacts avec le client et les utilisateurs et allant jusqu'à l'exploitation de la solution..

Contexte Général du Projet

1.1 Etat de l'art

Les différents types de systèmes d'information sont exposés à plusieurs menaces de sécurité et d'attaques qui peuvent largement influencer ou nuire le fonctionnement normal d'un système, et parfois dans certaines situations, même cesser complètement son fonctionnement. Par conséquent, la sécurité des systèmes d'information est devenue une discipline importante, et une préoccupation qui doit être prise en compte dès les premières étapes de développement du système [1]

Dans les approches d'ingénierie des exigences classiques, la sécurité est considérée uniquement comme une caractéristique ou une exigence non-fonctionnelle. Les exigences de sécurité dans ces approches traditionnelles, sont modélisées comme des contraintes de qualité qui doivent être prises en considération dans l'étape d'implémentation avec d'autres exigences non-fonctionnelles, telles que la disponibilité, la fiabilité et la performance [2]. Les différents mécanismes de sécurité sont donc intégrés dans une conception existante qui peut ne pas être en mesure de les prendre en considération. De plus, les exigences de sécurité peuvent générer des conflits avec les exigences fonctionnelles du système quand ils ne sont pas pris en compte dans un stade précoce.

La conception d'un système sécurisé constitue une étape importante et délicate pour les développeurs de logiciels. Cette étape doit se baser sur une approche ou une méthodologie d'ingénierie systématique. Dans ce contexte, l'ingénierie des exigences de sécurité est une étape clé et un processus central dans le cycle de vie du développement de systèmes/logiciels qui suit une philosophie de sécurité par conception. Elle permet aux développeurs de systèmes d'information d'identifier les menaces et les risques de sécurité, leurs conséquences et les contre-mesures avant que le système ne soit mis en place, plutôt que comme une réaction à des attaques potentiellement désastreuses [3]. Les exigences de sécurité sont utilisées ensuite pour contrôler le développement et le déploiement de

logiciels.

L'ingénierie des exigences de sécurité représente une tâche particulièrement complexe. Les exigences de sécurité peuvent être difficiles à identifier, à exprimer et à gérer car les concepteurs ne doivent pas seulement prendre en compte les logiciels en cours de conception, mais également les interactions entre les personnes, les organisations, le matériel et logiciels [4]. Au cours de ces dernières décennies, les chercheurs et les ingénieurs ont développé un nombre considérable de techniques d'ingénierie des exigences de sécurité, certains d'entre eux ont déjà gagné une grande popularité comme les arbres d'attaque. SysML est exemple de ce type de technique, il a été défini comme une extension d'un sous-ensemble du langage UML (Unified Modeling Language).

Dans le cadre de ce projet, nous sommes particulièrement intéressés par une extension du langage SysML appelée SysML-Sec, introduite par Apvrille et Roudier dans [5]. SysML-Sec est un environnement permettant de concevoir des systèmes embarqués sûrs et sécurisés ciblant les composants logiciels et matériels de ces systèmes. Cet environnement est entièrement pris en charge l'outil TTool (prononcé "tea-tool") qui permet d'éditer des diagrammes UML et SysML, ainsi que d'utiliser des outils de simulation et de vérification formelle intégrés afin d'analyser des propriétés de sécurité, de sûreté et de performance dans ces diagrammes

Une autre discipline d'ingénierie des exigences appelées "l'ingénierie des exigences de sûreté" nous intéresse aussi dans le contexte de ce projet. En effet, les deux disciplines de sûreté et de sécurité se sont développées comme deux disciplines distinctes pendant de nombreuses années, menées par des communautés développant chacune leurs propres outils et méthodologies [6][7]. Néanmoins, même si l'ingénierie des exigences de sûreté et de sécurité sont des problématiques différentes et elle ne doivent pas être fusionnées ou traiter de la même façon, elles sont également étroitement liées et partagent de nombreux points communs et des exigences similaires. Des définitions non absolues de sûreté et de sécurité sont données dans [2] comme suit. La sécurité est liée à des risques provenant ou exacerbés par une intention malveillante, indépendamment de la nature de la conséquence connexe, alors que la sûreté concerne les accidents, c'est-à-dire sans intention malveillante, mais avec des impacts potentiels sur l'environnement du système.

1.2 Etude et familiarisation

L'objectif de cette étude est de savoir comment utiliser la notation SysML dans le cadre d'un processus complet partant des premiers contacts avec le client et les utilisateurs et allant jusqu'à l'exploitation de la solution.

SysML est un langage de modélisation graphique développé par l'OMG, INCOSE

et AP233. Il ré-utilise une bonne partie des diagrammes UML et il fournit aux ingénieurs un langage de modélisation allant bien au delà des problématiques de l'informatique.

SysML est l'ingénierie des systèmes complexes ce qu'UML est à l'informatique. Donc l'objectif de SysML est de permettre à des acteurs de corps de métiers différents de collaborer autour d'un modèle commun pour définir un système. Généralement la conception de système génère souvent une accumulation de documentations qui doivent toutes être croisées et mises à jour pour maintenir et assurer la cohérence et respecter l'ensemble des spécifications du système. Cependant, SysML représente le moyen pour regrouper dans un seul modèle commun à tous les corps de métiers, les spécifications, les contraintes et les paramètres de l'ensemble du système.

SysML n'aborde plus la conception avec la notion de classes mais avec la notion de blocs qui présentent ou bien qui deviendront des parties électroniques, mécaniques, informatique ou autres, figure 1.1, 1.2, 1.3.

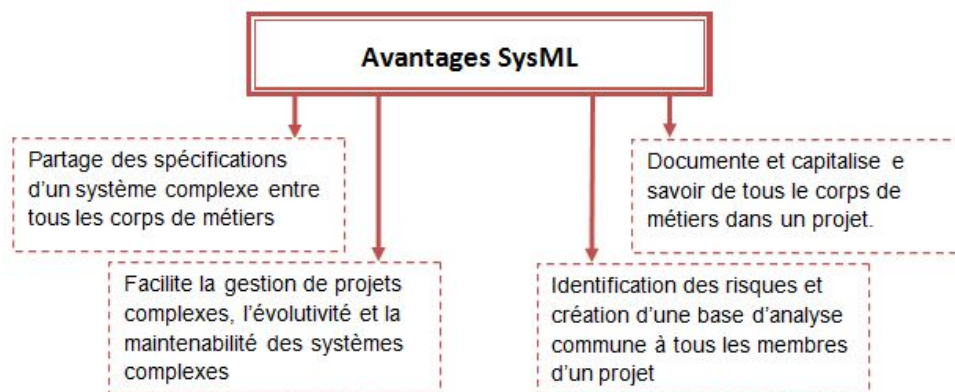


FIGURE 1.1 – Avantages SysML

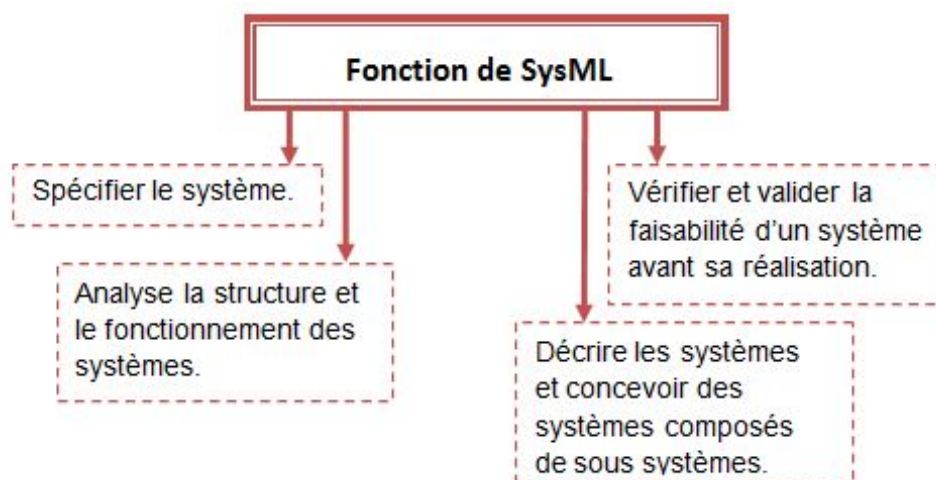


FIGURE 1.2 – Fonctionnalités de SysML

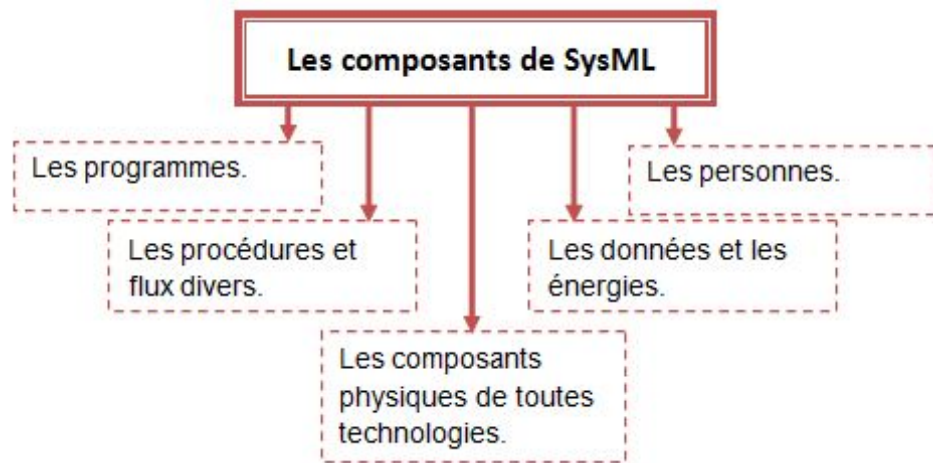


FIGURE 1.3 – Les composants de SysML

1.3 Comparaison entre SysML et UML

En SysML chaque diagramme est nommé d'une façon précise et il constitue un élément du modèle, figure 1.4. Pour cela SysML définit une en-tête à chaque diagramme qui contient obligatoirement, figure

- Le type de diagramme : bdd, act, ibd, ... ;
- Les éléments représentés dans le diagramme : package, blocs, activités, ... ;
- Le nom de l'élément modélisé ;
- Le nom du diagramme ou de la vue représentée.

De plus, chaque diagramme dispose d'une description :

- Version ;
- Niveau d'avancement ;
- Description ;
- Référence.

La principale différence entre UML et SysML réside dans le fait qu'UML utilise des « Class » quand à SysML utilise des « Block ».

- **Le diagramme paramétrique**(Figure 1.6 :)

Le « Parametric diagram » est utilisé pour exprimer les contraintes physiques entre les blocs avec des équations et des paramètres qui permettent de simuler le fonctionnement du système.

Le diagramme paramétrique peut ensuite être utilisé pour faire des simulations qui vérifieront si le système répond ou non aux spécifications.

Exemple : Machine à café (Figure 1.7).

<u>SysML</u>	<u>Description</u>	<u>UML</u>
Use Case diagram	Identique en UML et en SysML, il modélise les fonctionnalités que le système doit fournir. Le cas d'utilisation est une unité fonctionnelle utilisée pour la description du système.	Use Case diagram
Sequence diagram	Identique en UML et en SysML le diagramme de séquence modélise la chronologie des interactions entre les éléments du système ou entre le système et l'extérieur.	Sequence diagram
Activity diagram	Même utilisation en UML et en SysML. Le diagramme d'activité modélise les flux d'informations et les flux d'activité du système.	Activity diagram
State Machine diagram	Identique en UML et en SysML, il représente les différents états que peut prendre un élément ou une opération ainsi que ses réactions aux événements extérieurs.	State Machine diagram
Block Definition diagram	Le diagramme de Bloc en SysML est semblable au diagramme de Classe en UML. Il donne une représentation statique des entités du système, de leurs propriétés, de leurs opérations et de leurs opérations.	Class diagram
Internal Block diagram	Le diagramme interne de bloc SysML et le diagramme composite UML donnent une représentation « Boîte blanche » qui matérialise les imbrications des parties et leurs interconnexions par les ports.	Composite Structure diagram
Package diagram	Le diagramme de Package montre l'organisation générale du modèle en UML comme en SysML. En SysML il sert en plus à donner différentes vues du système.	Package diagram
Parametric diagram	Nouveau dans SysML ce diagramme modélise les paramètres physiques du système. Il sert à tester les performances physiques et quantitatives du système avant la réalisation et le déploiement.	None
Requirement diagram	Le diagramme de spécification nommé aussi diagramme d'exigence est nouveau dans SysML et il permet de collecter et d'organiser toutes les exigences textuelles du système.	None
Allocation tables	Nouveau en SysML. Les tables d'allocation sont de simples tableaux et non des diagrammes qui récapitulent les spécifications afin de faciliter le suivi de projet.	None

FIGURE 1.4 – Comparaison : UML et SysML

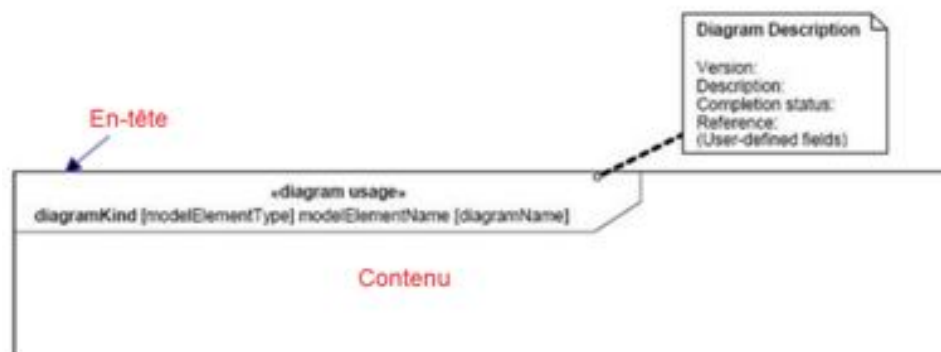


FIGURE 1.5 – Vue d'en-tête d'un diagramme SysML

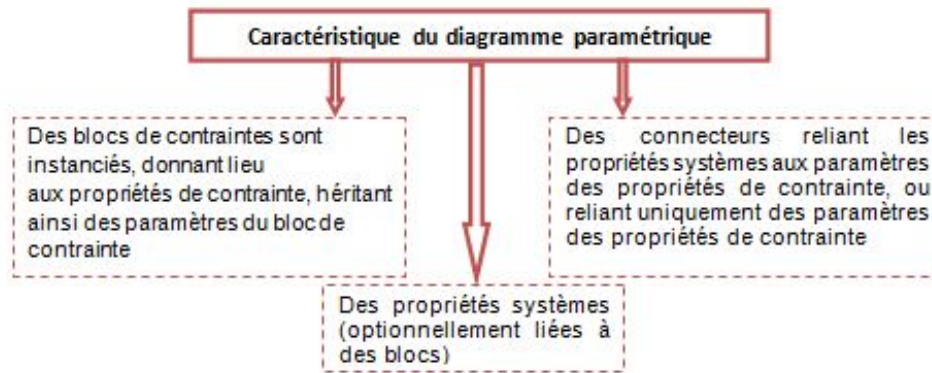


FIGURE 1.6 – Caractéristique du diagramme paramétrique

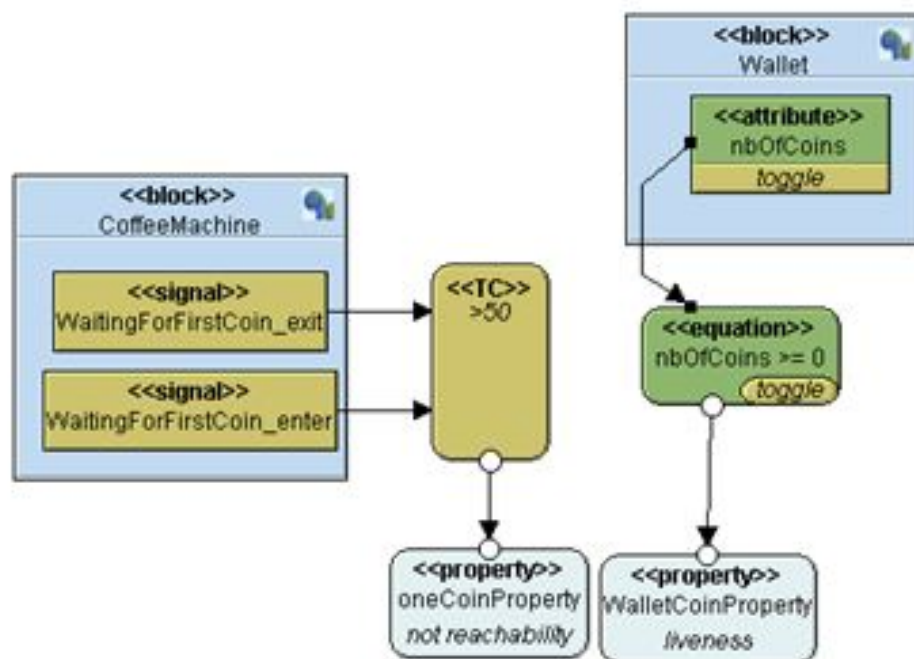


FIGURE 1.7 – Diagramme paramétrique

Ce diagramme représente mathématiquement les paramètres de la machine à café. Par exemple le seuil pour entrer ou sortir de la monnaie. Ainsi, les règles qui décrivent l'évolution et le changement des paramètres.

- **Le diagramme d'exigence** (Figure

Ce diagramme est utilisé pour collecter toutes les exigences techniques, physiques, commerciales ou autres dans un projet.

Ce diagramme est transversal à l'intégralité du système et il permet en plus de hiérarchiser les spécifications.

La définition des exigences nous permet donc d'élaborer le cahier de charge.

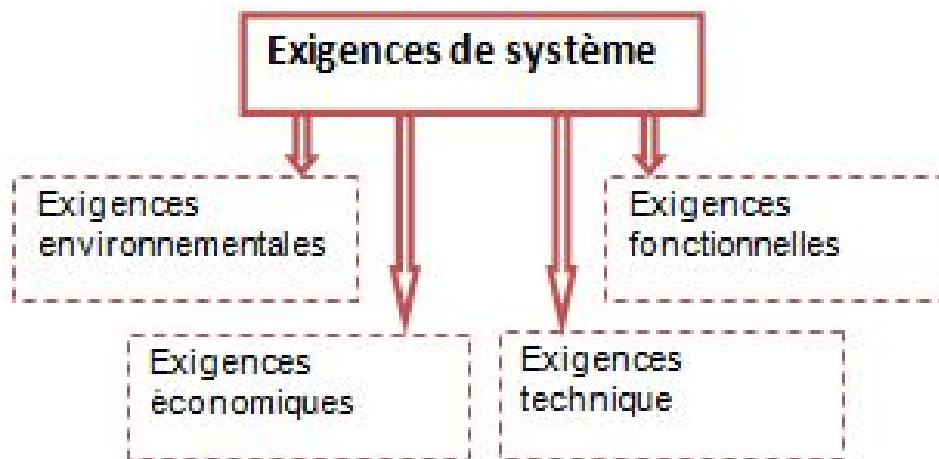


FIGURE 1.8 – Diagramme d'exigence

Chaque exigence est décrite par un texte à l'intérieur d'un rectangle marqué d'un stéréotype «requirement» identifié de façon unique. Ces exigences peuvent être reliées par des relations de contenance, de raffinement ou de dérivation.

Exemple : Machine à café

On commence par classer les exigences en représentant les blocs avec des attributs.

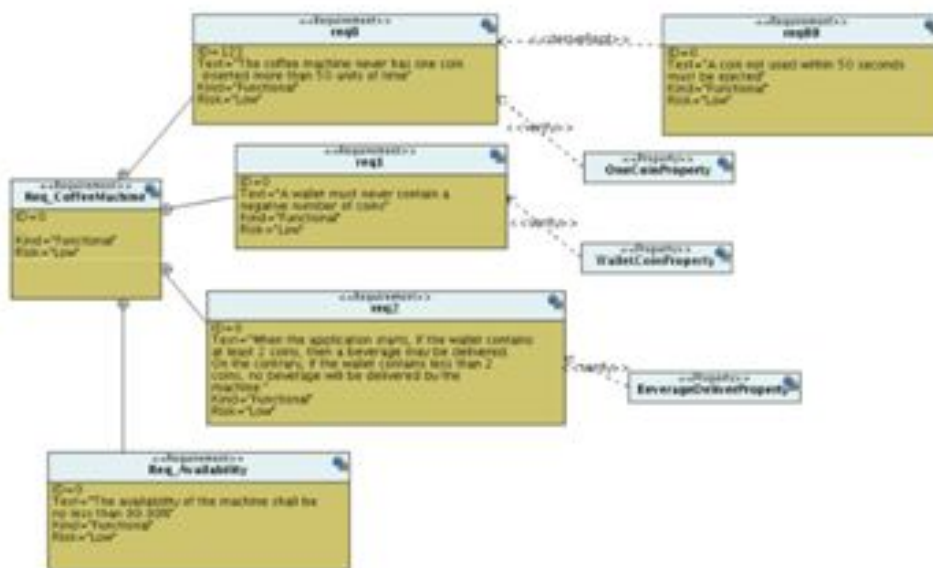


FIGURE 1.9 – Diagramme d'exigence, Machine à café

Les exigences de ce système sont (Figure

- Le temps de garder la monnaie et de choisir ;
- La machine doit toujours contenir de la monnaie ;
- La condition de servir une boisson ;
- La disponibilité de la machine.

1.4 SysML-sec

SysML-Sec est un environnement permettant de concevoir des systèmes sûrs et SÉCURISÉS avec une version étendue du langage SysML. SysML-Sec cible les composants logiciels et matériels de ces systèmes.

Les diagrammes SysML-sec (SysML) sont édités en utilisant l'outil open source Ttool. Cet outil permet ainsi de faire la simulation et à la vérification formelle (sécurité, sûreté, performance) de ces diagrammes.

On utilise l'ensemble de ces diagrammes afin d'avoir un système fonctionnel et bien sécurisé. Le partitionnement du système, les exigences de sécurité et les menaces sont progressivement affinés. Cette opération se fait donc en suivant les étapes dans la figure 1.10 qui commence par une première architecture et en itérant jusqu'à atteindre un résultat satisfaisant :

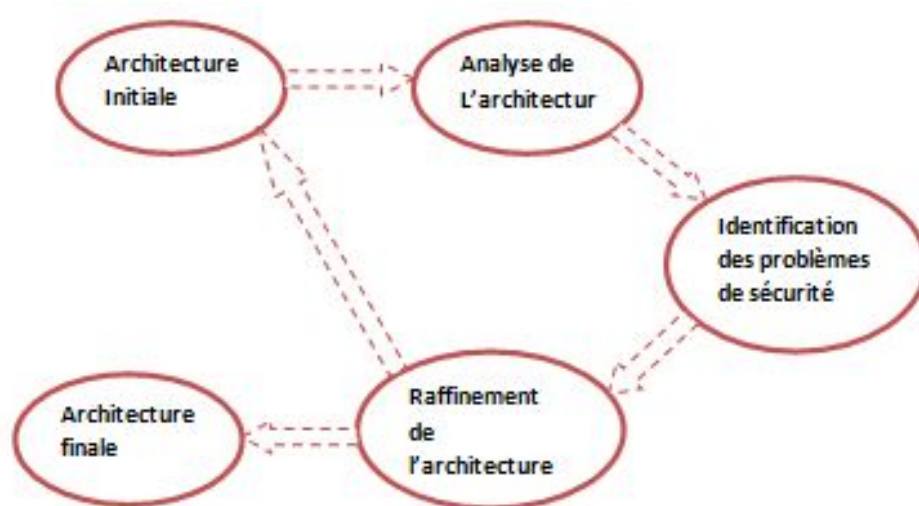


FIGURE 1.10 – Etapes pour avoir une architecture valide

1.5 L'outil TTool

L'outil Ttool est un logiciel open-source dédiée à la création des diagrammes UML et SysML ainsi que la simulation et la vérification formelle des propriétés de sécurité et de performance dans ces diagrammes (Figure 1.11).

- **DIPLODOCUS** : Profil UML dédié au partitionnement de systèmes ;
- **AVATAR** : Environnement basé sur SysML pour décrire une structure arborescente d'exigences exprimées en langage naturel ;

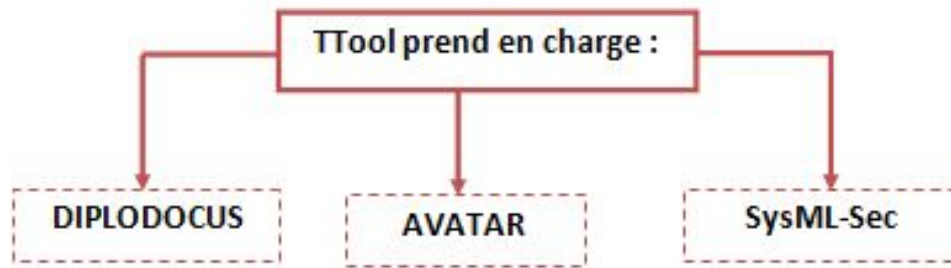


FIGURE 1.11 – Définition TTool

- **SysML-Sec** : c'est l'extension qui nous intéresse comme cité auparavant. C'est un environnement permettant de concevoir des systèmes sûrs et sécurisés avec une version étendue du langage SysML et il cible les composants logiciels et matériels de ces systèmes.

TTool permet de réaliser un certain nombre de fonctionnalités parmi lesquelles on peut citer : La vérification formelle des diagrammes UML, la possibilité d'effectuer des vérifications des exigences temporelles et d'appliquer une vérification formelle à une analyse basée sur un cas d'utilisation et un scénario (une personne qui ne connaît pas la conception orientée objet peut utiliser TTool).

Etude du projet

2.1 Objectif

L'objectif principal de notre projet consiste à étendre l'outil TTool en développant un front-end plus portable pour permettre aux concepteurs d'accéder à l'outil via une interface Web. Ce front-end doit prendre en considération la création des différents types de diagramme UML, SysML et SysML-sec.

- **Front-end pour TTool** : Création d'une application web qui permet aux concepteurs de bénéficier des fonctionnalités de TTool avec la possibilité de collaboration et de partage ;
- **Inclusion des outils d'analyse des diagrammes** : Prendre en considération des différents back-end disponible dans l'outil TTool permettant de vérifier les propriétés de sécurité du système modélisé avec l'outil.

On devait commencer tout d'abord par sélectionner l'ensemble des technologies qu'on peut utiliser choisir les outils avec lesquels on va travailler. On a deux choix :

- Application web JEE basée sur les APIs servlet ;
- Application web basée sur javascript, html et css.

Après une étude globale de ces technologies, on a choisi de travailler avec la première méthode qui est l'application web JEE basée sur les APIs servlet Parcequ'on se basant sur l'application desktop TTool on ne pourra pas passé directement vers une application web.

2.2 Problème rencontrés

Durant la réalisation de notre projet, on a rencontré plusieurs problèmes parmi lesquels on cite :

- **Manque de documentation sur l'utilisation de l'outil TTool :**

Puisque notre travail consiste à relier entre l'application TTool et l'application qu'on va créer, on devait tout d'abord commencer par :

- Comprendre toutes les fonctionnalités de TTool ;
- Comment choisir le type des diagrammes à dessiner ;
- Comment décrire un diagramme ;
- Comment relier entre les diagrammes ;
- Comment valider et vérifier les diagrammes.

L'outil TTool ne permet pas aux utilisateurs d'effectuer l'ensemble de ces options facilement parce que l'interface n'est pas si claire et n'affiche pas tout les éléments qui permettent de dessiner les diagrammes.

La figure 2.1 représente la première fenêtre de l'application TTool :

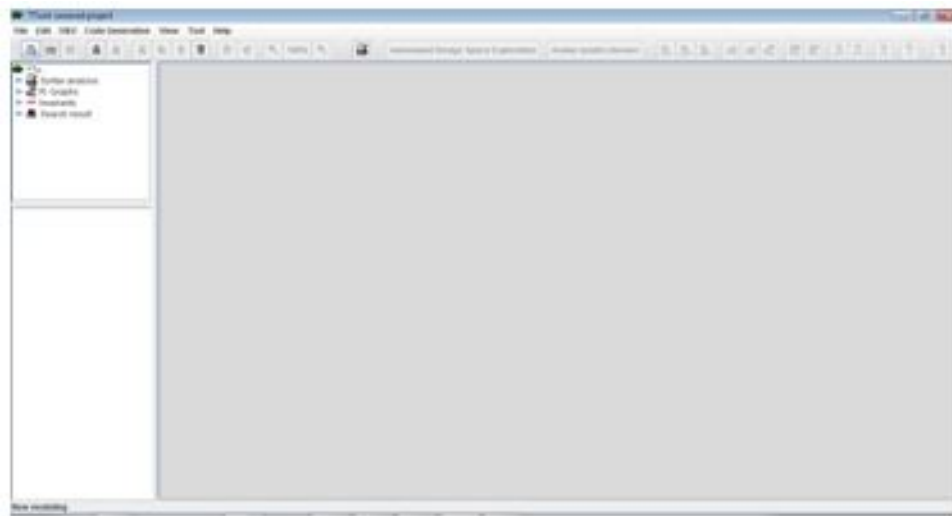


FIGURE 2.1 – La première interface de TTool

Pour choisir le type de diagramme qu'on veut dessiner, il faut cliquer sur le bouton droit, figure 2.2 :

Une fois on choisit le type de diagramme, ça nous permettra de d'afficher les composants qu'on pourra utiliser pour le dessiner, figure 2.3.

Manque de commentaire et d'explications du code source

Pour pouvoir relier entre le back end qui est TTool et le front end qu'on va créer,

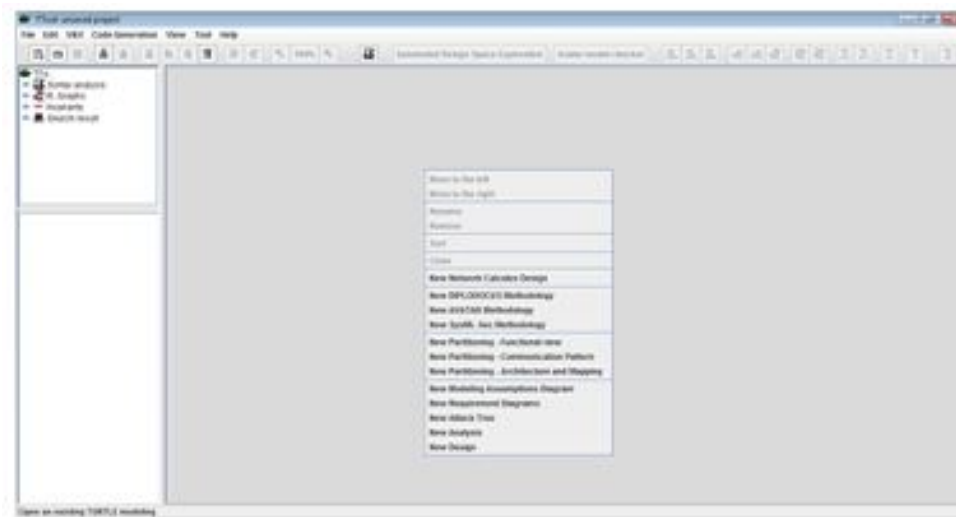


FIGURE 2.2 – Choix du type de diagramme

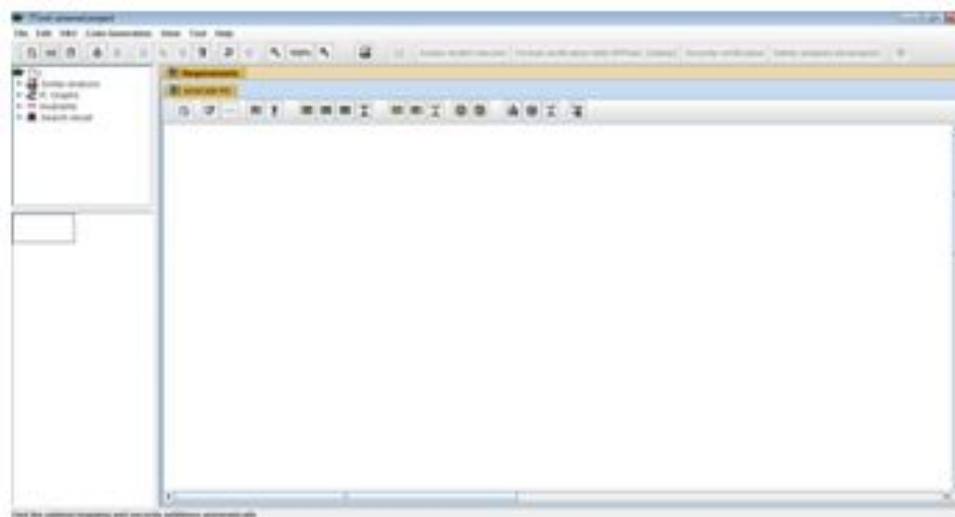


FIGURE 2.3 – Interface pour dessiner les diagrammes

on doit comprendre l'architecture de TTool chose qui n'était pas expliqué ni documenté, figure 2.4 .



FIGURE 2.4 – Les composants du code source de TTool

Réalisation

3.1 Environnement de développement

3.1.1 Environnement matériel

L'équipement mis à notre disposition pour la réalisation du projet se compose d'un ordinateur portable dont la configuration est la suivante :

- Processeur : Intel ® celeron ® CPU N3060 ;
- 4 GO de mémoire ;
- 500 GO d'espace disque.

3.1.2 Environnement logiciel

Après avoir présenté les moyens matériels mis à notre disposition dans le cadre de réalisation de ce projet, nous abordons dans cette partie les moyens logiciels utilisés.

Les logiciels utilisés pour la réalisation de ce projet sont :

- **Netbeans 8.1 :**

Un environnement de développement intégré (EDI), Il permet également de supporter différents autres langages, comme JavaScript, C, Java.



- **Javascript :**

Un langage de programmation de script, Il est principalement employé dans les pages WEB interactives. Il permet de rendre l’affichage dynamique.



- **HTML :**

Un langage qui a pour rôle de gérer et organiser le contenu d’une page web. C’est un langage de description de données, et non un langage de programmation. Nous avons utilisé le HTML5 qui est la dernière version de HTML.



- **CSS :**

Un langage permettant de gérer l’apparence des pages web (agencement, positionnement, décoration, couleurs, taille du texte...). C’est un langage complément du langage HTML pour obtenir une page web avec du style. Le navigateur parcourt le document HTML. Lorsqu’il rencontre une balise, il demande au CSS de quelle manière il doit l’afficher.



- **JQuery :**

Une librairie JavaScript qui permet de faciliter les fonctionnalités de ce langage. L'utilisation de cette librairie permet de gagner du temps lors de la programmation JavaScript.



- **Bootstrap :**

Un framework HTML, CSS et JavaScript créé par des développeurs de Twitter en 2010. Il permet de créer facilement de design d'un site tout en assurant que celui-ci soit responsive, c'est-à-dire qu'il puisse s'adapter automatiquement à toutes les formats d'écran (Tablettes, ordinateurs, appareil mobile etc.)



- **MxGraph :**

Dans notre projet, nous avons utilisé la bibliothèque MxGraph, qui est une API basé sur JavaScript qui permet de dessiner des graphes dans des applications Web.



- **L'architecture MVC :**

MVC est un sigle pour "Modèle, Vue, Contrôleur" en français, et c'est une technique de conception fréquemment employée dans les projets basés sur la programmation orienté objets, comme c'est le cas pour notre projet. Cette technique a comme but de faciliter la modularité, la flexibilité et l'utilisation des composants programmés, elle permet aussi d'associer plusieurs vues distinctes à un même modèle. En plus, un changement au niveau de l'implémentation d'un modèle (optimisation, réorganisation, ...) n'impliquerait pas forcément des changements au niveau des vues et des contrôleurs. Cela va conduire en pratique à la séparation des objets en une séquence d'interaction particulière en trois catégories. Ces trois parties sont :

- Le modèle qui est la partie décrivant les données à afficher (Dans notre projet, les fonctionnalités de TTool) ;
- La vue qui est la représentation graphique de ces données, et c'est ce composant qui représente les écouteurs du modèle et qui est averti du changement des données ;
- Le contrôleur qui est la partie qui traite des interactions du composant avec l'utilisateur.

La figure ci-dessous illustre le principe de fonctionnement :

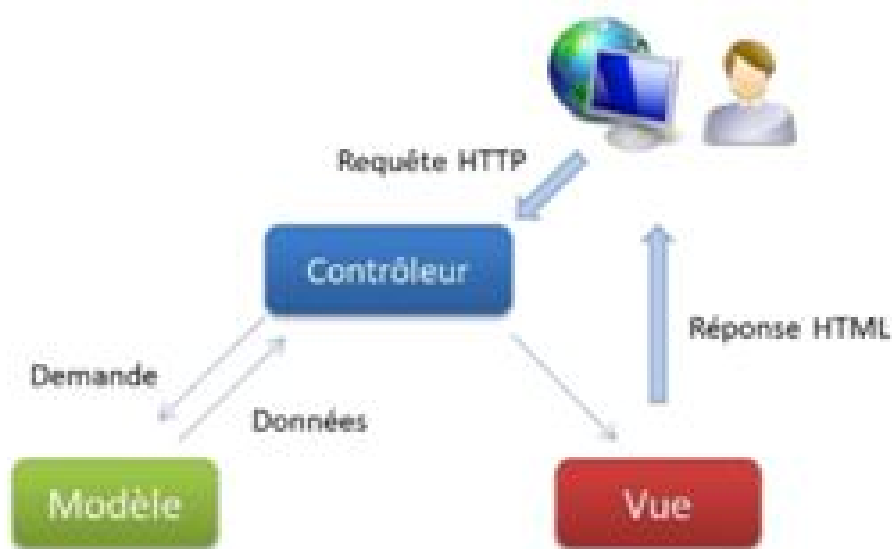


FIGURE 3.1 – Schéma du Modèle, Vue et Contrôleur [8]

Le MVC fonctionne comme suit :

- Le client exécute une requête sur le serveur ;
- Le contrôleur récupère la requête généralement par le biais de dispatcher. Le contrôleur va se charger de toutes les demandes et les calculs ;
- Le contrôleur récupère la requête généralement par le biais de dispatcher. Le contrôleur va se charger de toutes les demandes et les calculs ;
- Le modèle va jouer le rôle de surcouche logicielle pour la base de données (Dans notre application, fichiers XML) ;
- Quand le contrôleur a fini d'exécuter son code, il fait appel à la vue. La vue va uniquement se charger de l'affichage et de la mise en forme de la donnée.

- **Plateforme JEE :**

Java Enterprise Edition ou Java EE, est une spécification pour la technologie Java de Sun plus particulièrement destinée aux applications d'entreprise. Dans ce but, toute implémentation de cette spécification contient un ensemble d'extensions au Framework Java Standard (JSP, Java Standard Edition) afin de faciliter la création des applications réparties.



Prototypage

4.1 Maquettage

Avant d'entamer la phase de développement, on a créé des maquettes qui représentent l'interface qu'on compte avoir, et cela en utilisant l'outil balsamiq qui est un logiciel rapide et très intuitif, ce logiciel nous permet de poser une structure pour toutes les pages de notre application, grâce à sa bibliothèque d'éléments très fournie. Il jouit d'une bonne réputation dans la communauté des designers d'interfaces et UX designer.

Notre application contient cinq onglets :

- Le premier onglet contient la page d'accueil de notre application et il décrit le contenu de notre application. Figure
- Le deuxième onglet représente l'éditeur qui nous permet de dessiner les diagrammes SysML figure
- Le troisième onglet contient un guide d'utilisateurs, il facilite l'utilisation de l'outil en expliquant l'ensemble des fonctionnalités ;
- Le quatrième onglet contient une documentation sur l'ensemble des diagrammes UML, SysML et SysML-sec ;
- Le cinquième et dernier onglet fournit les informations sur les créateurs de l'application.

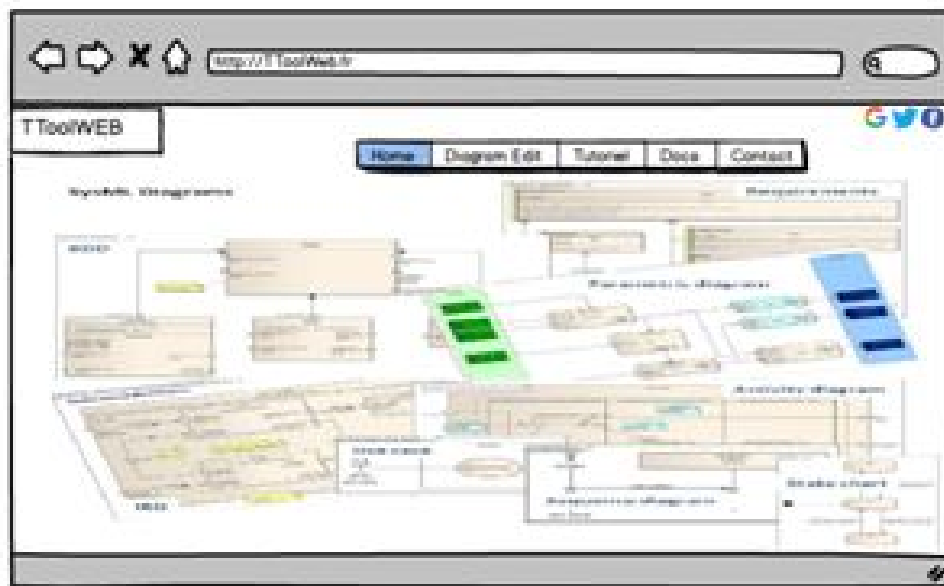


FIGURE 4.1 – Maquette de la page d'accueil de l'application

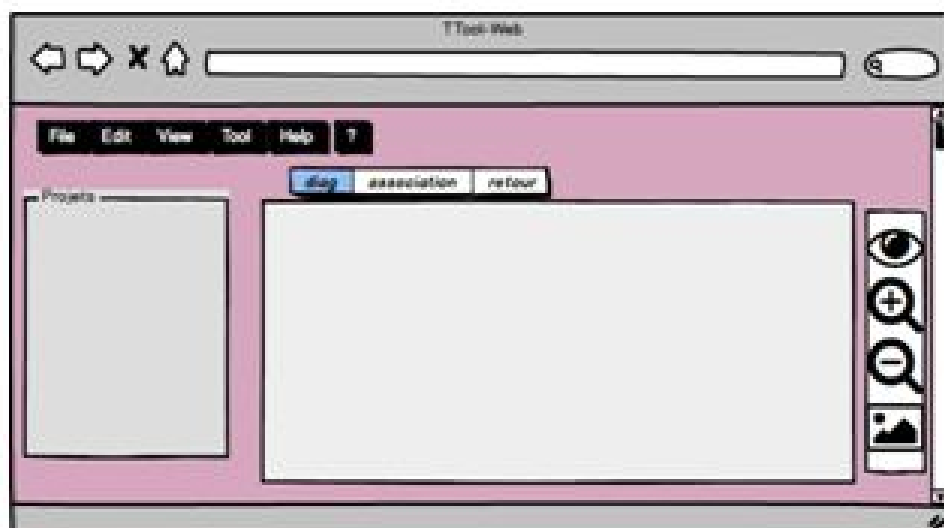


FIGURE 4.2 – La maquette de l'éditeur des diagrammes

Démonstration

5.1 Notre application

L'application qu'on a crée finalement contient quatre onglets, figure 5.1.

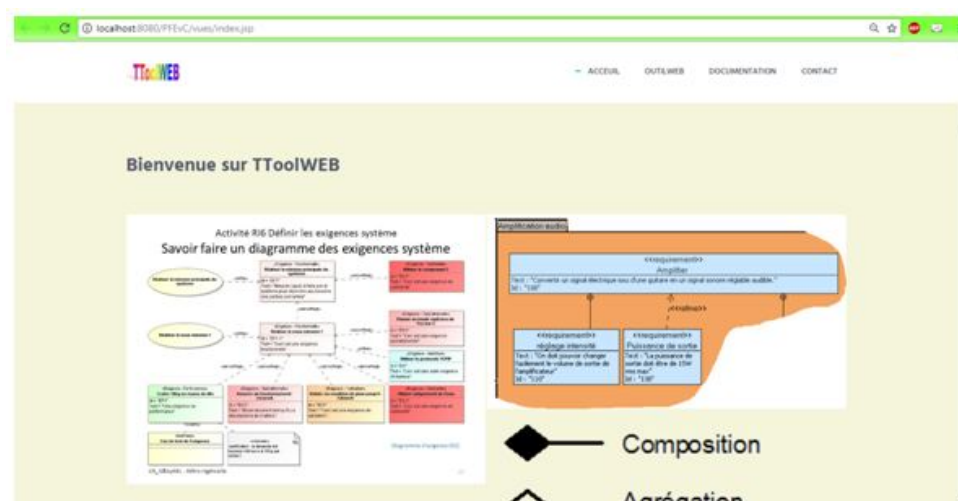


FIGURE 5.1 – Première interface de TToolWEB

Cette page contient une représentation générale des diagrammes UML, SysML et SysML-sec.

Ensuite le deuxième onglet « OutilWeb » nous permet d'accéder à l'outil TToolWEB, figure

Une fois on lance l'outil, on va avoir la figure

Cette interface affiche l'ensemble des diagrammes supportés pas TToolWEB ainsi que l'ensemble de ses composants. Il nous permet de modifier la police, couleur et taille des éléments qu'on dessine ainsi que le canvas ou on va dessiner. Cet outil permet aux utilisateurs d'importer ou exporter des modèles existantes sous format XML.



FIGURE 5.2 – Onglet de l'éditeur TToolWEB

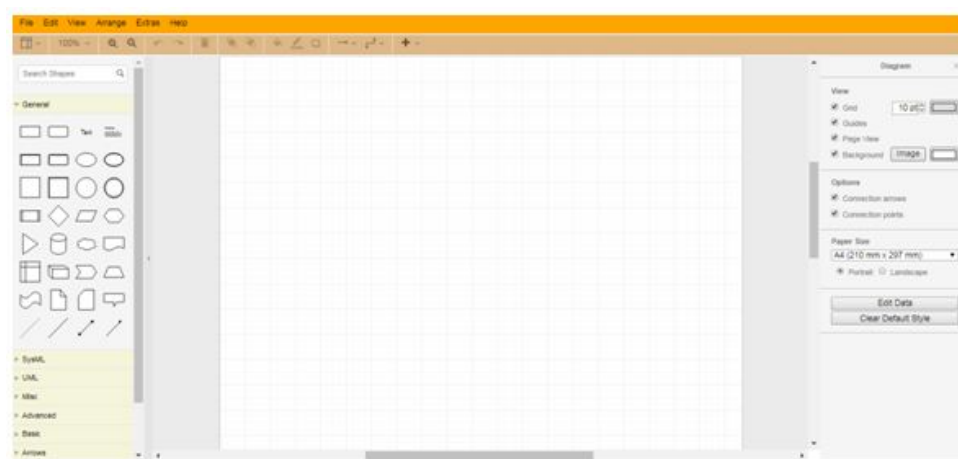


FIGURE 5.3 – L'outil TToolWEB

Le troisième onglet « Documentation » permet à l'utilisateur d'exploiter et de découvrir le fonctionnement de TToolWEB, figure 5.4.

Et finalement le dernier onglet contient nos coordonnées afin de rester en contact avec les utilisateurs, figure 5.5

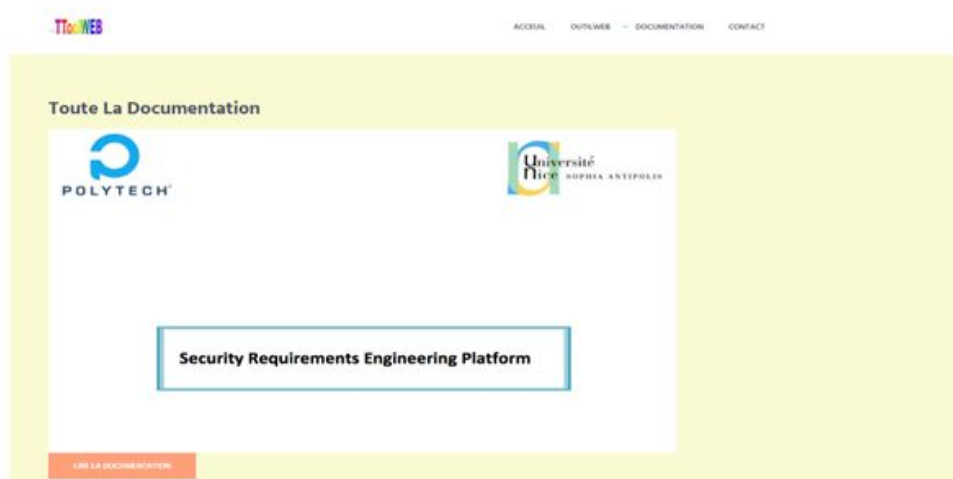


FIGURE 5.4 – Onglet Documentation

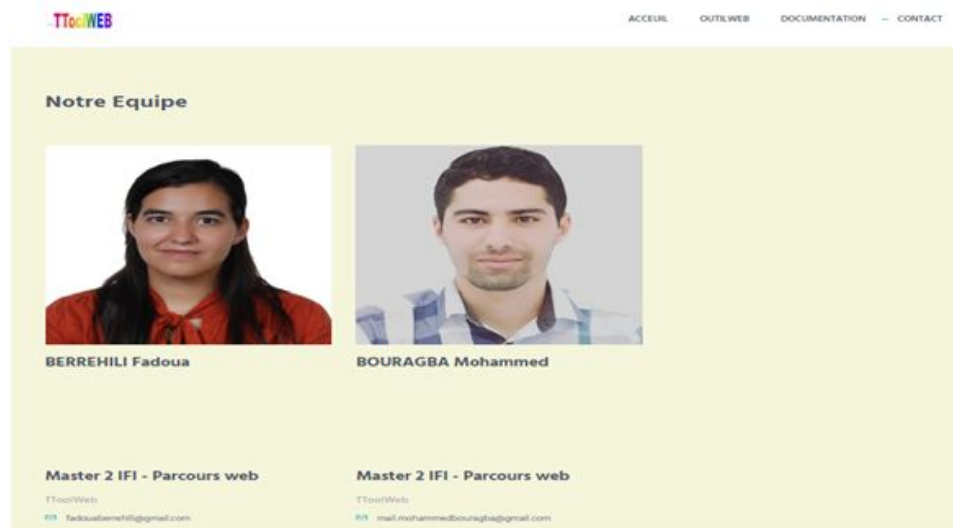


FIGURE 5.5 – Onglet Contact

5.2 Composition de notre application

La figure 5.6 représente comment on a utilisé les technologies cités dans le paragraphe auparavant dans notre application.

La partie VIEWS contient les pages JSP créées en HTML5, CSS MxGraph et Javascript ainsi par Bootstrap. Ils communiquent avec les modèles qui contiennent les fonctionnalités de l'application TTool desktop en utilisant les servlets.

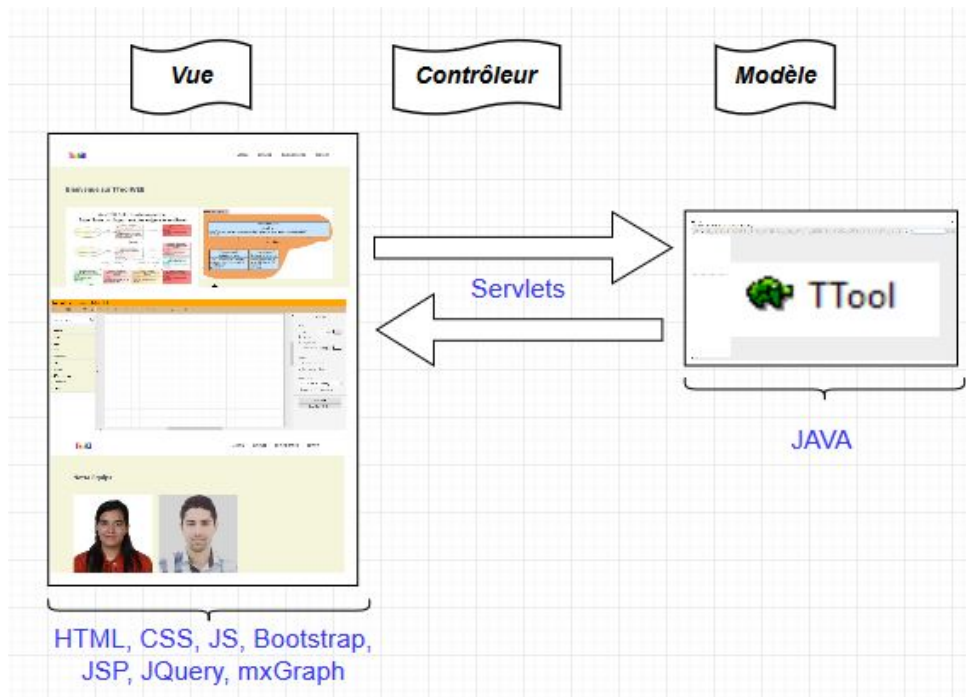


FIGURE 5.6 – Composition de notre application

On a choisit l'architecture MVC (modèle, vue et contrôleur) parceque c'est un concept très puissant qui intervient dans la réalisation d'une application ce qui est notre cas. Son principal intérêt est la séparation des données (modèle), de l'affichage (vue) et des actions (contrôleur).

Cette approche MVC (figure 5.6) apporte de réels avantages parmi lesquels :

- Une conception **claire** et **efficace** grâce à la séparation des données de la vue et du contrôleur ;
- Un **gain de temps** de maintenance et d'évolution du site ;
- Une plus **grande souplesse** pour organiser le développement du site entre différents développeurs (indépendance des données, de l'affichage (webdesign) et des actions).

Dans notre application chacun de ces éléments est représenté comme suit :

La vue :

Elle correspond à un fichier HTML, CSS, JSP, Bootstrap, MxGraph et JQuery. Elle se contente d'afficher le contenu qu'elle reçoit sans avoir connaissance des données.

Le Contrôleur :

C'est l'élément qui va utiliser les données pour les envoyer à la vue. Son rôle est donc de récupérer les informations, de les traiter en fonction des paramètres demandés par la vue, puis de renvoyer à la vue les données afin d'être affichées.

Le Modèle :

Le modèle correspond aux fonctions de TTool.

Cette architecture MVC qu'on a utilisé est une solution très intéressante et très puissante, à mettre en œuvre dans de grands projets de préférence.

L'indépendance des trois couches est très importante, car c'est grâce à cette différenciation que l'architecture puise sa puissance et sa logique.

Il est ainsi facile de mettre à jour le design de son site, sans toucher aux données ni à la façon dont elles sont organisées. Inversement, il est aussi facile de modifier la structure de stockage des données ou la manière dont elles sont gérées sans bouleverser l'affichage.

Ensuite pour l'éditeur des graphes, on s'est basé sur la bibliothèque MxGraph qui est un composant JavaScript qui fournit des fonctionnalités destinées aux applications qui affichent des diagrammes et des graphiques interactifs. MxGraph fournit quelques fonctionnalités requises pour dessiner, interagir et associer un contexte à un diagramme. Elle est livrée avec un certain nombre d'exemples qui aident à expliquer comment une application de base est mise en place et présente les caractéristiques individuelles de la bibliothèque.

Conclusion Générale

En guise de conclusion, il convient de rappeler que ce travail n'est qu'une tentative d'améliorer l'outil TTool qui est un outil de modélisation des diagrammes SysML.

Nous avons exposé dans ce rapport un état de l'art sur les différents diagrammes et méthodes de conception. Nous avons ensuite exposé notre démarche qui consiste dans le fait de créer une application web et aussi lier entre les fonctionnalités de TTool et l'application qu'on a créée.

Ce projet nous a permis d'appliquer les connaissances qu'on a acquises au cours de nos années de formation. On a rencontré de nombreux problèmes et obstacles auxquels on a tenté de trouver des solutions et des alternatives. En effet, ce projet nous a offert l'opportunité de découvrir et d'utiliser des outils et des moyens auxquels nous n'avions pas accès auparavant.

Le contact avec les personnes qui nous devancent en matière de développement, n'a pas seulement un apport technique mais aussi humain, ouvrant des portes vers toutes les qualités qu'on doit posséder, en matière de savoir être avant le savoir-faire, pour s'offrir toutes les conditions propices au développement humain et professionnel, chose qu'on a apprécié et essayé de mettre en pratique.

Bien que nous avons effectué des recherches approfondies pour faire l'amélioration de TTool et pouvoir l'étendre en y accédant via le web . D'autres investigations peuvent être menées dans la problématique que nous avons abordée.

Nous proposons comme amélioration l'ajout de l'option de partage des données entre les utilisateurs, chose qui permettra à chaque utilisateur d'enregistrer ses données dans le cloud et de pouvoir y accéder à distance, et cela en respectant la synchronisation d'accès aux modèles.

Enfin, on espère, avoir participé humblement au développement de cet outil par ce travail qui est le fruit de l'aide et du soutien de la direction et principalement de nos enseignants qui nous ont prodigués des recommandations et des conseils précieux.

Bibliographie

- [1] A. Souag, R. Mazo, C. Salinesi and I. Comyn-Wattiau (2015). “Reusable knowledge in security requirements engineering : a systematic mapping study”. *Requirements Engineering*, vol. 21, pp. 251-283.
- [2] L. Piètre-Cambacédès and M. Bouissou (2013). “Cross-fertilization between safety and security engineering”. In *Reliability Engineering System Safety*, vol. 110, pp. 110-126.
- [3] N. Mayer (2012). “Model-based Management of Information System Security Risk”. Presses universitaires de Namur.
- [4] A. Souag (2012). Vers une nouvelle génération de définition des exigences de sécurité fondée sur l’utilisation des ontologies. *INFORSID 2012*, May 2012, Montpellier, France, pp.583-590.
- [5] L. Apvrille and Y. Roudier (2013). “SysML-sec : A sysML environment for the design and development of secure embedded systems”. In : *APCOSEC 2013*, Asia-Pacif Council on Systems Engineering, Yokohama, Japan. Yokohama, JAPAN (09 2013).
- [6] E. Schoitsch (2004). “Design for safety and security of complex embedded systems : a unified approach”. In : *Proceedings of the NATO Advanced Research Workshop on Cyberspace Security and Defense*, Gdansk, Poland, pp. 161–174.
- [7] M. B. Line, O. Nordland, L. Røstad and I. A. Tøndel (2006). “Safety vs security?”. In : *Proceedings of the 8th International Conference on Probabilistic Safety Assessment and Management (PSAM 2006)*, New Orleans, Louisiana, USA, 2006.
- [8] <https://openclassrooms.com/courses/apprenez-a-programmer-en-java/mieux-structurer-son-code-le-pattern-mvc>

RAPPORT DU PROJET DE FIN D'ÉTUDES

MASTER INFORMATIQUE, FONDAMENT ET INGÉNIERIE

Parcours

WEB

Security Requirements Engineering Platform

Melle Fadoua BERREHILI
Mr Mohammed BOURAGBA

Sous la direction de :
M. Yves ROUDIER

Membres de jury :
M. Michel RIVEILL
M. Molines GUILHEM

Le 26 janvier 2018