Security Requirements Engineering Platform

Réalisation :

- **BERREHILI Fadoua (fadouaberrehili@gmail.com)**
- **BOURAGBA Mohammed (mail.mohammedbouragba@gmail.com)**
- **SAHLI Hamza (sh.hamza.90@gmail.com)**

Master 2 IFI- Parcours web

2017/2018

Contexte (Hamza)

1. Description

De nos jours, 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.

2. Objectifs

Les objectifs principaux du ce projet PFE sont les suivants :

- **Front-end basé JavaScript pour TTool.** L'objectif principal consiste à étendre l'outil TTool en développant un front-end plus portable basé sur JavaScript pour permettre aux concepteurs d'accéder à l'outil via une interface Web. Ce front-end ou interface doit prendre en considération la conception de différents types de diagramme de SysML-sec.

- **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.

- **État de l'art**. Effectuer une étude bibliographique académique sur les travaux existants qui traitent la problématique d'interaction entre les exigences de sécurité et de sûreté.

- **Extension de SysML-Sec.** La proposition d'une extension de l'approche SysML-sec pour permettre de capturer des interactions entre les exigences de sécurité et de sûreté et de prendre en charge de leur modélisation et leur analyse en se basant sur une étude bibliographique. Cela devrait aboutir à l'extension des diagrammes SysML-sec et du front-end TToo lpour les prendre en charge. Il devrait également conduire à l'élaboration d'un outil d'analyse pour identifier et analyser ces interactions.

- **Autres extensions possibles de l'outils TTool.** L'ajoute d'une couche de collaboration pour permettre à plusieurs développeurs de coopérer et de participer à la spécification des exigences de sécurité et de sécurité. Aussi la révision du format de sauvegarde des données et des diagrammes disponible actuellement.

Etude et familiarisation- SysML- (Fadoua)

1. Description

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.

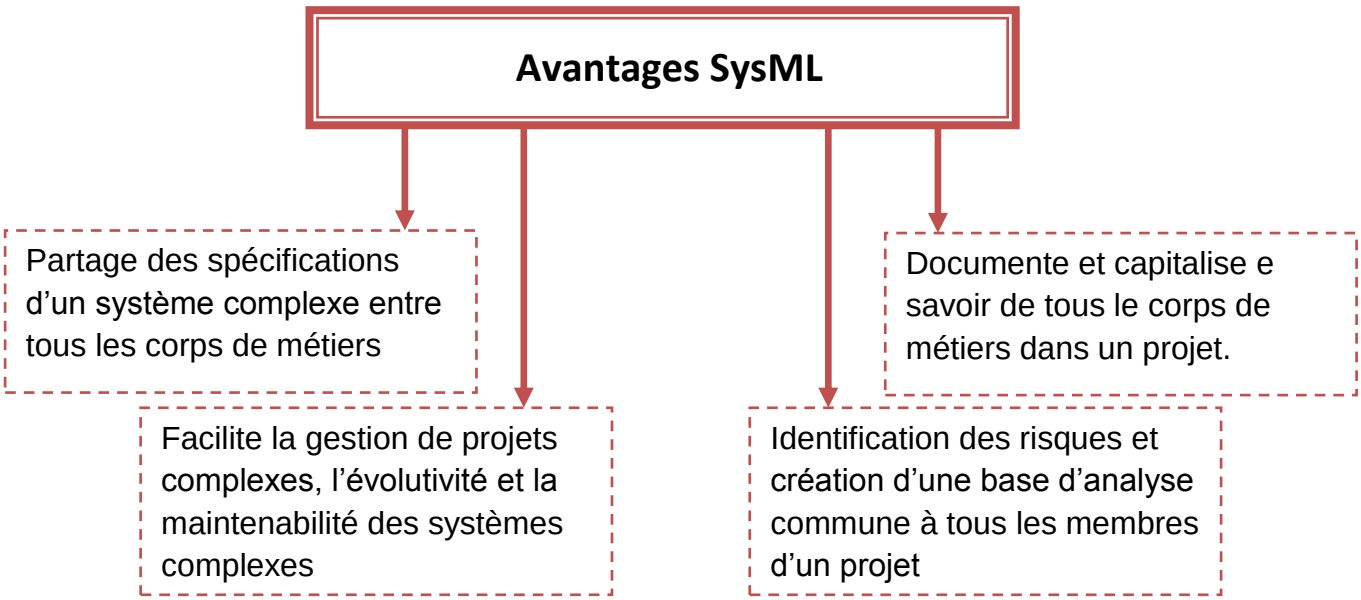

Figure 1 : Avantages SysML.

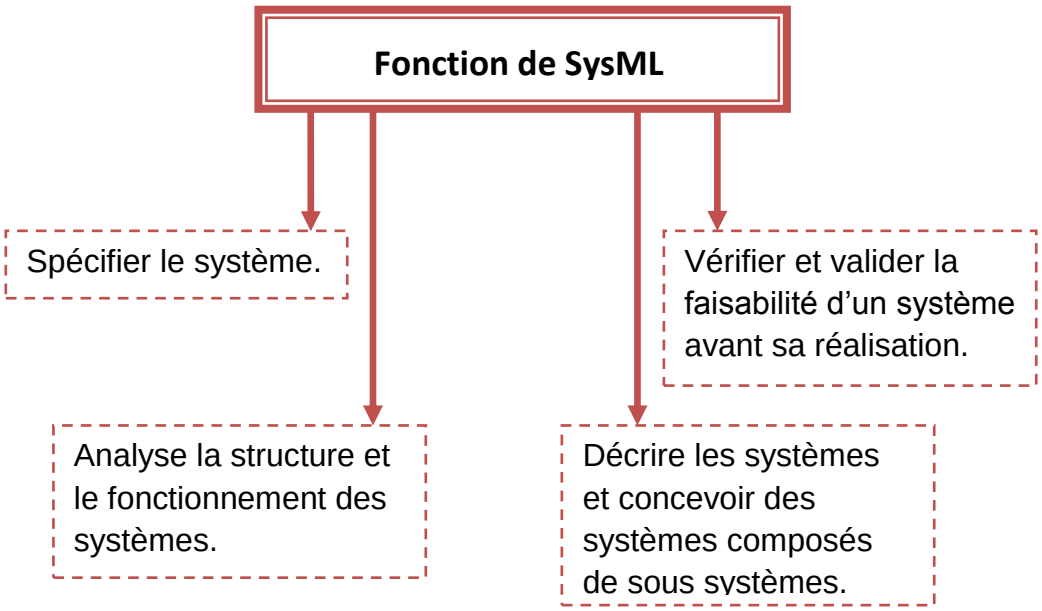

Figure 2 : Fonction SysML.

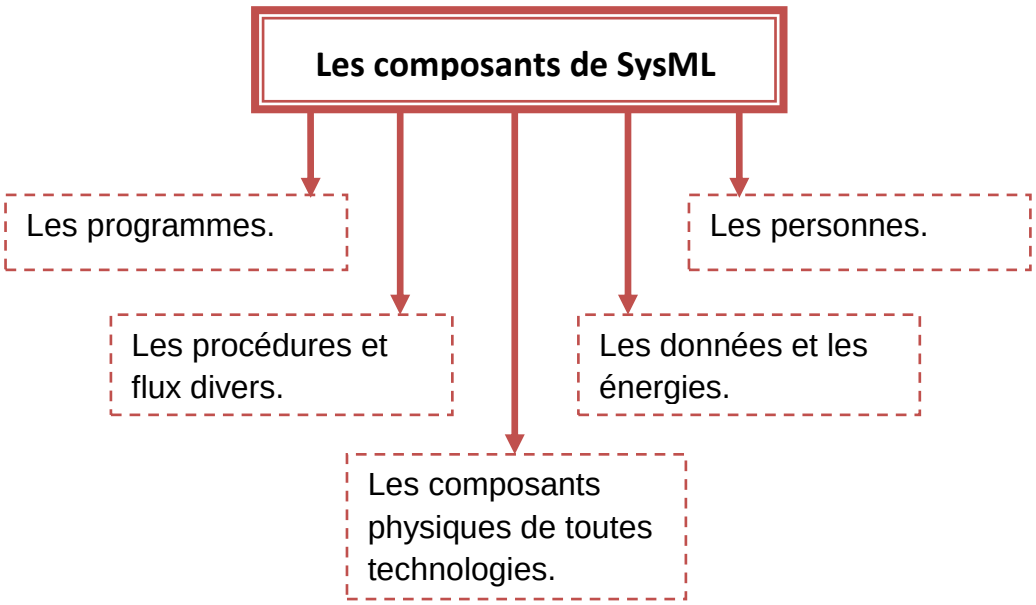

Figure 3 : Les composants SysML.

2. Comparaison entre UML et SysML :

SysML	Description	UML
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 évènements 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 internet de bloc SysML et le diagramme composite UML donnent une représentation « Boîte blanche » qui matérialise les imbrications des parties et leurs interconnections 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**

Tableau 1:Comparaison: UML et SysML.

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

- 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.

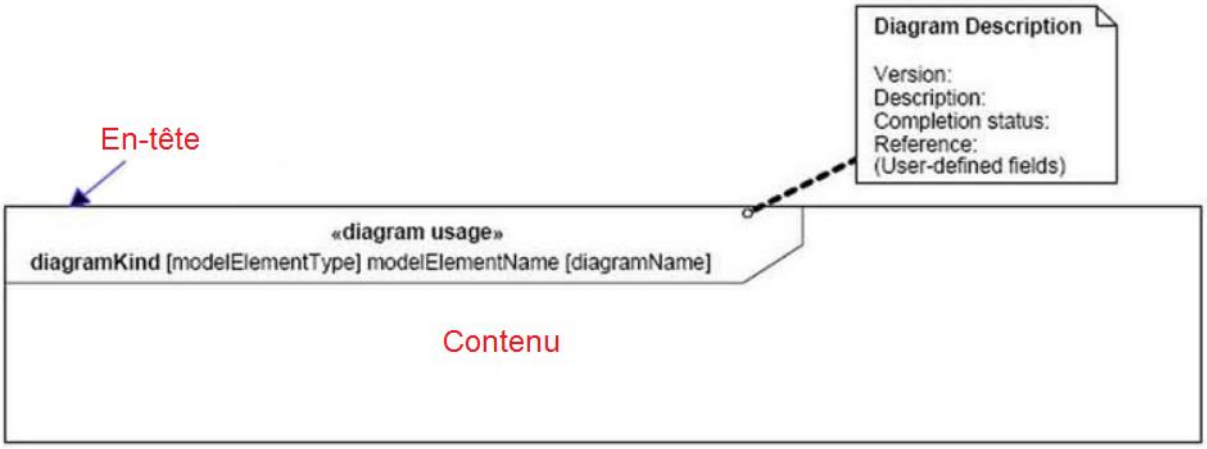

Figure 4 : Vue d'en-tête d'un diagramme SysML.

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 :

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.

Le diagramme d'exigence :

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.

3. 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 fonctionnelle et bien sécurisé. Le partitionnement du système, les exigences de sécurité et les menaces sont progressivement affinés. Cette opération se fait donc on suivant les étapes dans la figure ci-dessous qui commence par une première architecture et en itérant jusqu'à atteindre un résultat satisfaisant:

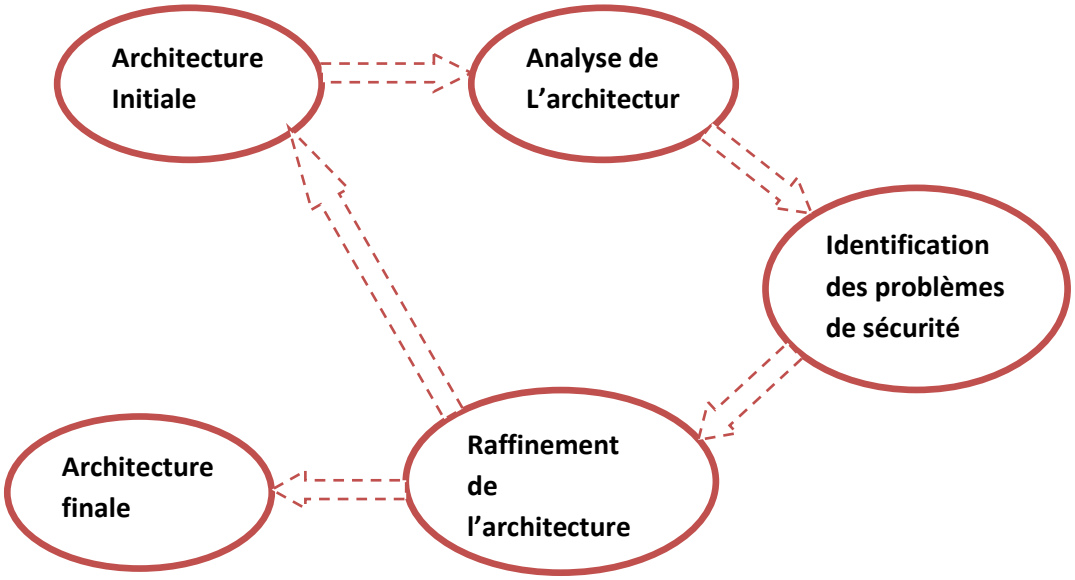

Figure 5 : Etapes pour avoir une architecture valide.

4. Création d'une première maquette :

Notre objectif global du projet est de créer une application (Front end) qui permet aux concepteurs de créer les diagrammes SysML en se basant sur TTool.

Comme première étape, on a fait une maquette qui représente la page d'accueil de cette application et qui va nous permettre d'accéder à d'autres pages.

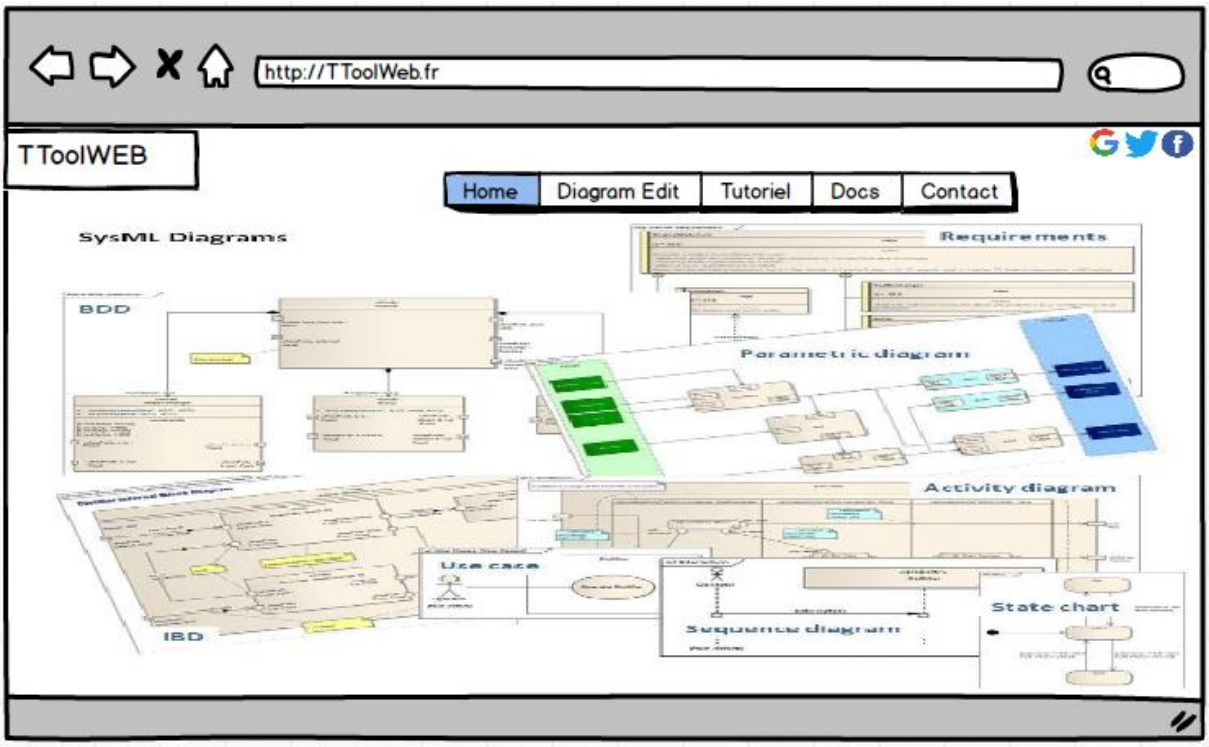

Figure 6 : Maquette de la page d'accueil.

Cette page va nous permettre d'accéder à la page présentée dans la figure X qui va nous permettre de créer des diagrammes SysML.

Démarche et étapes à suivre : (Mohammed)

1. Description

Après la compréhension de notre problématique, on voulait 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 deux méthodes, on a choisi de travailler avec la deuxième méthode qui est l'application web basée sur javascript. En s'inspirant du fonctionnement de l'outil Ttool qui 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.

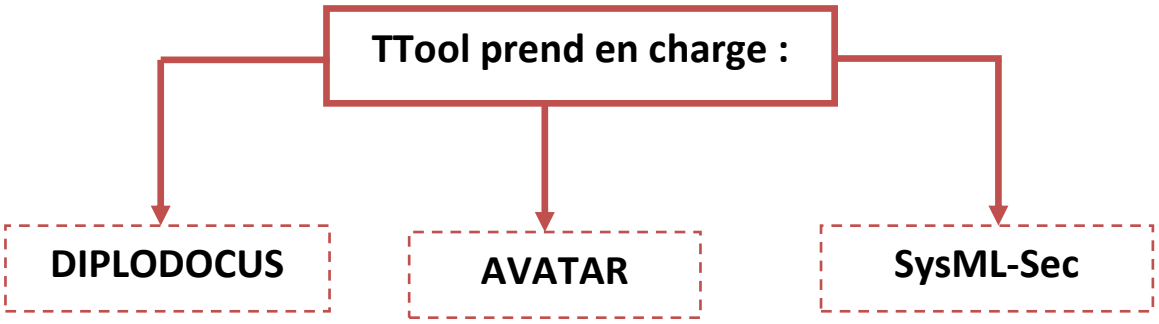

Figure 7 : Définition TTool.

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

On a tout d'abord commencé par créer une interface sur balsamiq pour avoir une idée sur le produit qu'on veut atteindre. La figure ci-dessous représente l'interface qu'on compte créer :

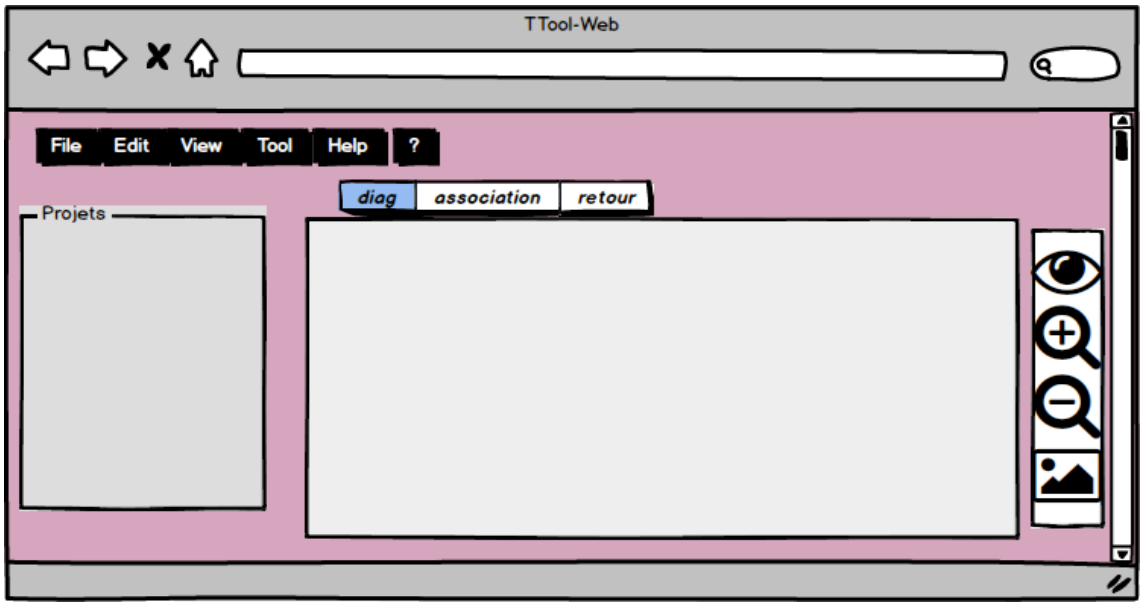

Figure 8 : Interface créée par balsamiq

Cette interface va permettre aux concepteurs de créer les concepts de modélisation, relations, attribut et aspects en dessinant des métamodèles basé sur les diagrammes SysML,

Dans notre cas on va se baser uniquement sur :

- Les diagrammes d'exigence : Il permet de spécifier une capacité ou une contrainte qui doit être satisfaite par un système. Elle peut spécifier une fonction que le système devra réaliser ou une condition de performance, de fiabilité, de sécurité, etc. Les exigences servent à établir un contrat entre le client et les réalisateurs du futur système.
- Les diagrammes paramétrique : Il permet de représenter des contraintes sur des valeurs de paramètres système, telles que performance, fiabilité, masse, et il s'agit d'une spécialisation du diagramme blocs internes ou les seuls blocs utilisables sont des contraintes entre les paramètres permettant de représenter graphiquement des équations et des relations mathématiques ;

Donc on doit commencer tout d'aborde par bien comprendre ces deux types de diagrammes pour pouvoir les utiliser.

1. **Diagramme des exigences :**

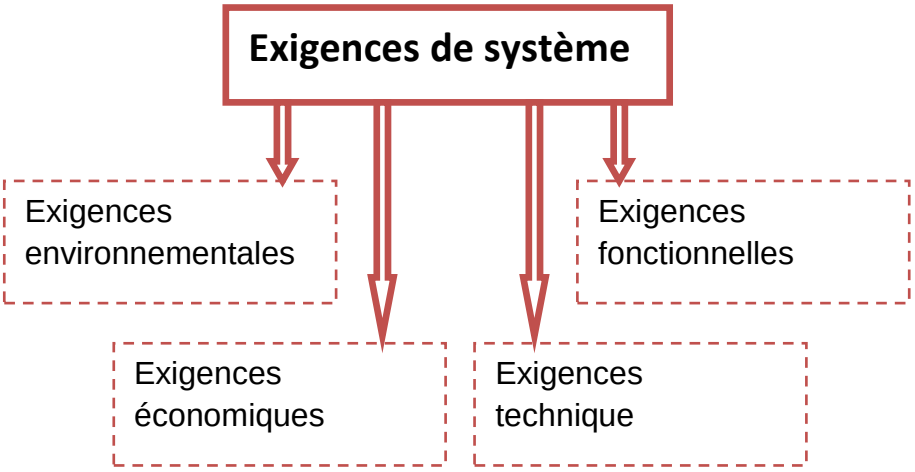

Figure 9 : Diagramme d'exigence

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

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 classifier les exigences en représentant les blocs avec des attributs.

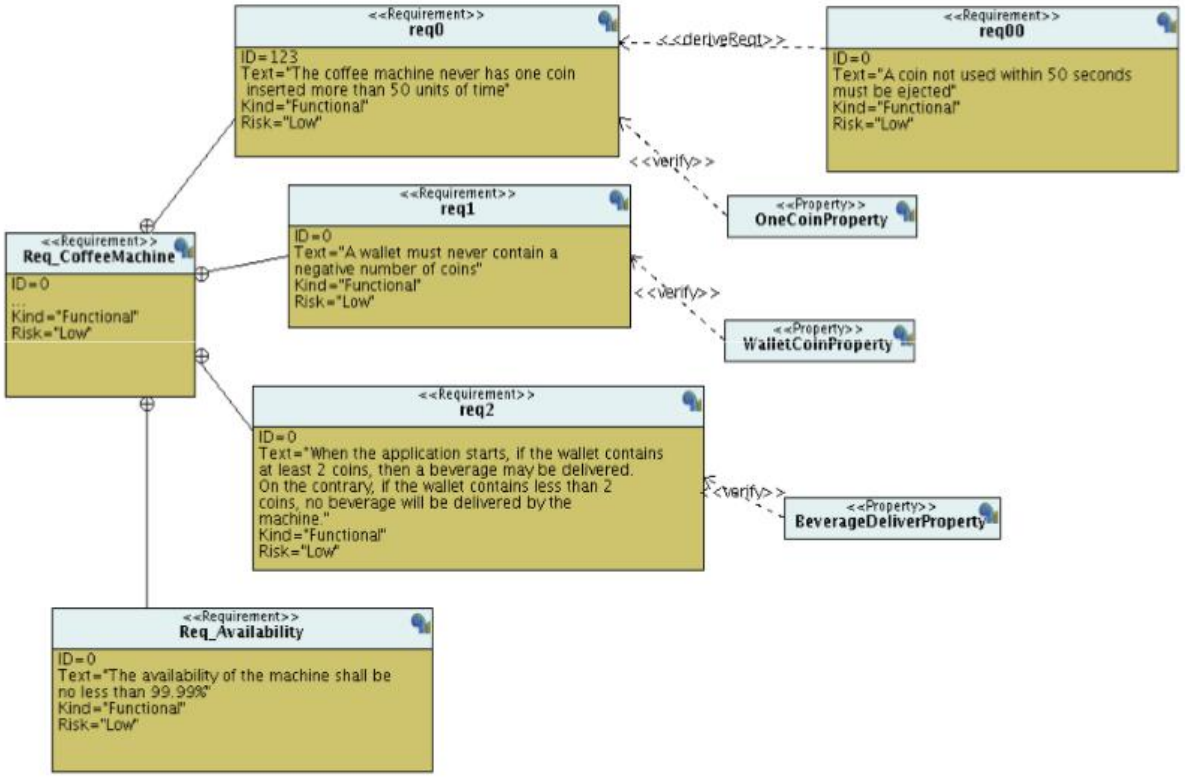

Figure 10 : Diagramme d'exigence, Machine à café.

Les exigences de ce système sont :

- 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.

2. **Diagramme paramétrique :**

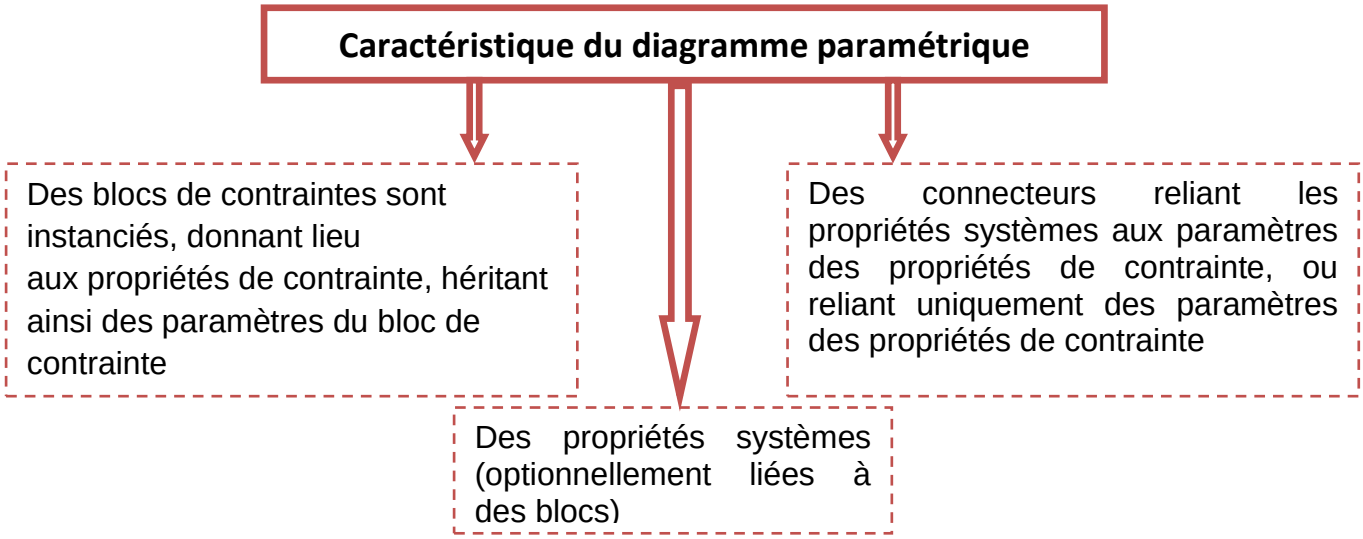

Figure 11 : Caractéristique du diagramme paramétrique

Exemple : Machine à café

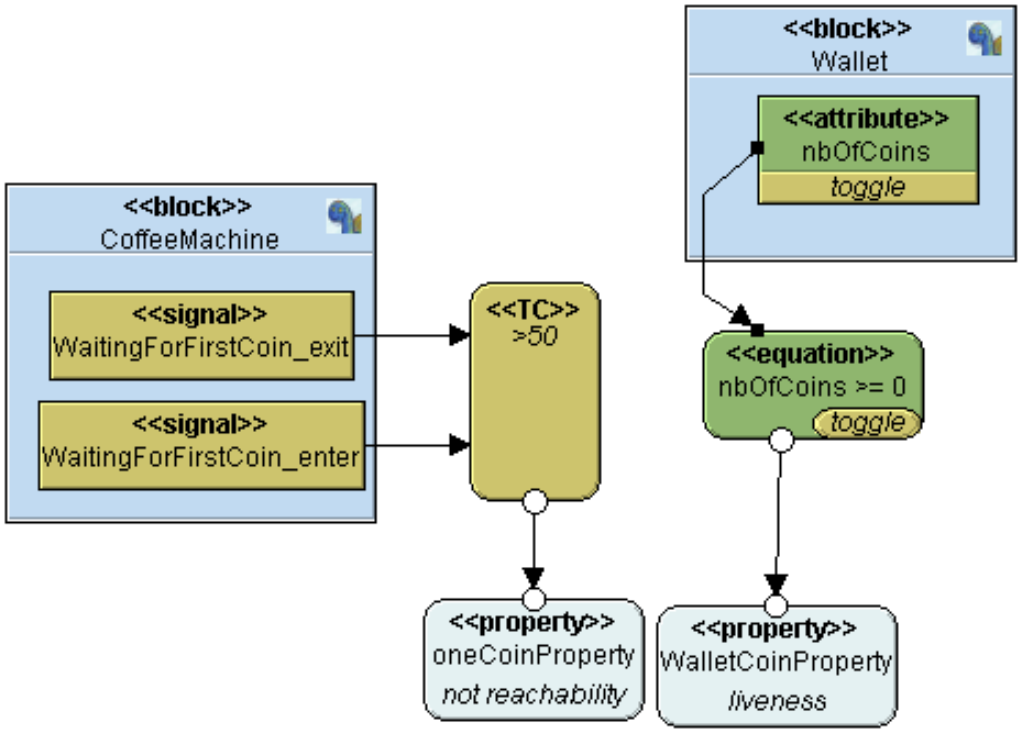

Figure 12 : Diagramme paramétrique

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 décrient l'évolution et le changement des paramètres.

Planning de la première semaine

Durant la première semaine, on doit avoir en résultat les interfaces implémentées de notre application et aussi une idée l'implémentation des diagrammes.

Mohammed : Implémentation de l'interface qui nous permettra de dessiner les diagrammes.

Fadoua : Implémentation de la page d'accueil de l'application web et ajout des fonctionnalités à l'interface créée par Mohammed.

Table de figure :

REFERENCES :

[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-Pacific 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 theNATO 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.