Analyse de la vulnérabilité humaine dans les systèmes de systèmes socio-techniques

Paul Perrotin

DOCTORAT BRETAGNE LOIRE / MATHSTIC

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

THÈSE DE DOCTORAT DE

L'ÉCOLE NATIONALE SUPÉRIEURE
MINES-TÉLÉCOM ATLANTIQUE BRETAGNE
PAYS-DE-LA-LOIRE - IMT ATLANTIQUE

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*

Par
Paul PERROTIN

Analyse de la vulnérabilité humaine dans les systèmes de systèmes socio-techniques

Thèse présentée et soutenue à l'École navale, Lanvéoc, le 14/12/2022
Unité de recherche : Lab-STICC UMR CNRS 6285
Thèse financée par et préparée au sein de la chaire de cyberdéfense des systèmes navals
Thèse N° : 2022IMTA0335

Rapporteurs avant soutenance :

Jean-Michel Bruel	Professeur des Universités, Université de Toulouse
Layth Sliman	Professeur des Universités, EFREI

Composition du Jury :

Président :	Christelle Urtado	Professeur, IMT Mines Alès
Examinateurs :	Jean-Michel Bruel	Professeur des Universités, Université de Toulouse
	Layth Sliman	Professeur des Universités, EFREI
Dir. de thèse :	Salah Sadou	Professeur des Universités, Université Bretagne Sud
	Antoine Beugnard	Professeur, IMT Atlantique Bretagne-Pays de la Loire
Enc. de thèse :	Nicolas Belloir	Maître de conférences, Académie Militaire de St-Cyr Coëtquidan

Invité(s) :

David Brosset	Maître de conférences HDR, Maître de conférences HDR, IRENAV, Arts et Métiers Sciences et Technologies
David Hairion	Architecte Systèmes Navals, Naval Group
Marc Pennamen	Responsable développement des offres et projets cybersécurité, Thales SIX

TABLE DES MATIÈRES

TABLE DES FIGURES

Remerciements

J'aimerais dans un premier temps remercier les personnes qui m'ont encadré lors de cette thèse : Nicolas Belloir mon encadrant, Salah Sadou et Antoine Beugnard, mes directeurs de thèse. Votre soutien tout au long de la thèse a été essentiel pour moi. Malgré les difficultés que nous avons rencontrées tout au long de la thèse, soit en ce qui concerne les cibles de publication ou même la crise sanitaire, vous avez été présents et m'avez guidé pour mener au mieux cette thèse à son terme. Durant ces quatre années, vous avez su canaliser les nombreux débats autour des thématiques de recherche, afin de me permettre de les mettre en œuvre dans cette thèse. En outre, vous m'avez énormément apporté sur le plan personnel. Pour tout cela, merci encore.

Merci aux rapporteurs de ce manuscrit, Jean-Michel Bruel et Layth Sliman qui, malgré un emploi du temps chargé en cette fin d'année 2022, avez pris le temps de vous investir dans la relecture de mon manuscrit. Vos remarques ont grandement contribué à améliorer sa qualité. Je tiens également à remercier l'ensemble des membres du jury pour le temps et l'intérêt que vous avez consacrés à mes travaux ainsi que pour la disponibilité dont vous avez fait preuve afin de venir sur le site à l'Ecole Navale assister à la soutenance malgré la contrainte que cela représente.

Je tiens à remercier chaleureusement David Hairion qui m'a accompagné durant toute la thèse. Merci pour ton temps et pour toutes les fois où tu nous as guidés sur les problématiques du monde de l'industrie. Merci aussi à toutes les personnes que j'ai pu rencontrer grâce à toi chez Naval Group. Le temps qu'ils m'ont consacré et le regard bienveillant qu'ils ont su apporter à mes travaux m'ont beaucoup aidé.

Bien sûr je ne saurai oublier dans ces remerciements la chaire de cyberdéfense des systèmes navals et toutes les personnes formidables que j'ai pu y rencontrer. Tout d'abord Yvon Kermarrec, directeur de la chaire : merci de m'avoir donné la chance de pouvoir faire cette thèse au sein de la chaire. David Brosset, coordinateur de la chair, merci pour ta présence au quotidien et pour ces nombreux cafés qui par moments s'étendaient plus que de raison, mais dans lesquels tu as toujours su donner une oreille attentive au terrible doctorant que j'étais. En parlant de terribles doctorants, il me faut remercier Douraid, Clet , Nicolas et Mael. Nous avons su former une sacrée équipe face à l'épreuve qu'est la

thèse et cela malgré des expériences quelque peu déroutantes à base de poisson fermenté et d'éléments japonais. Merci à Étienne, Bastien, Loubna et François pour tous ces moments que nous avons passés ensemble et toutes ces réunions dans notre second bureau qu'a été la Fabrik.

Je tiens aussi à remercier Olivier, Maxence, Jessica, Sébastien, Arthur, Benjamin, Mallorie et Perrine pour leur travail à la chaire et tous ces échanges que nous avons eus ensemble.

Je remercie Julien et Marc qui ont joué un rôle essentiel entre les doctorants et les industriels pour nous aider au mieux dans nos thèses.

J'ai une pensée particulière pour relève doctorale : Quentin, Robin et Erwan, je vous souhaite bon courage pour la thèse.

Guillaume et Ronan, merci pour ces week-ends où nous avons mangé plus que raison et ces soirées rennaises inoubliables. Cédric, merci pour toutes ces échappées à travers le jeu de rôle et toutes ces années d'amitié.

Ma chère Sophie, merci pour ton écoute ainsi que tes conseils qui m'ont été si précieux et pour tous les encouragements.

Enfin, je remercie ma famille qui a été là au quotidien pour moi : mon frère et mes sœurs ainsi que mon père. Je remercie particulièrement mon parrain Thierry pour m'avoir aidé à relire le manuscrit ainsi que pour toutes ces sorties où tu m'as fait découvrir tant de choses. Pour terminer, je souhaite particulièrement remercier ma mère : tu as consenti à tant de sacrifices pour que tes enfants réussissent, tu as été présente dans toutes les grandes épreuves de ma vie et notamment dans celle-ci, car sans toi je n'aurais pas pu aller aussi loin.

Première partie

Introduction

Chapitre 1

Introduction

Sommaire

Les termes *systèmes de systèmes socio-techniques* (SoSTS) sont complexes pour désigner quelque chose que nous côtoyons pourtant au quotidien. En effet les systèmes et organisations qui nous entourent ont gagné en complexité d'année en année. Cette complexité ne se traduit pas seulement par des systèmes unitaires plus complexes, mais aussi par des interactions entre ces systèmes pour générer plus de fonctionnalités. Prenons l'exemple d'un navire. Au début, dans sa plus simple expression, il s'agit d'un dispositif, souvent en bois, permettant de flotter et éventuellement de se mouvoir sur l'eau. Ce système simple est aujourd'hui largement enrichi par de nombreux éléments supplémentaires. Pour illustrer, prenons l'exemple d'un paquebot récent qui possède la capacité de communication, de lutte contre l'incendie, des cartes numériques, des déplacements géolocalisés, un pilotage de moteur, etc. Toutes ces nouvelles fonctionnalités et capacités ne sont pas de simples éléments techniques, ils sont en forte collaboration avec les personnes, ajoutant à cette complexité. Toute la dimension « socio-techniques » de ces systèmes de systèmes rentre ici en jeu. En effet les individus voulant utiliser ces systèmes doivent collaborer activement avec les éléments techniques et les personnes composant le système. Dans cette complexité toujours plus grande et ces fonctionnalités toujours plus nombreuses, est apparu une menace faisant partie intégrante de ce type de système : la menace cyber. En effet, plus les systèmes se sont connectés entre eux, plus cette menace est devenue importante et a explosé au fur et à mesure des années [1]. Aujourd'hui la menace se fait extrêmement pesante sur les systèmes de systèmes socio-techniques, d'autant plus que ces

SoSTS sont devenus critiques pour nos sociétés. L'exemple des hôpitaux est emblématique, car ceux-ci ont en effet subi en 2020 un nombre record d'attaques [1] mettant parfois en jeu la vie de patients [2]. Dans cette menace cyber, outre le côté technique, le vecteur principal d'agression sur les systèmes modernes est l'opérateur humain. En effet, les rapports récents [2] montrent que la majorité des attaques réussies ne sont pas le fait d'une vulnérabilité établie dans un système technique, mais bien la conséquence d'une action souvent malencontreuse menée par un opérateur humain qui a permis la réussite de l'attaque par négligence ou par méconnaissance des bonnes pratiques en matière informatique.

1.1 Chaire de cyberdéfense des systèmes navals

La chaire de cyberdéfense système navale a été créée en 2014 pour répondre aux nouveaux enjeux sur la cybersécurité du monde maritime. En effet, les actes de cybermalveillance sont en augmentation, et ce dans tous les secteurs, le secteur maritime n'y faisant pas exception. Aussi, pour les besoins du monde maritime, qu'il soit civil ou militaire, la recherche en cybersécurité des systèmes navals est aujourd'hui essentielle. Pour répondre à la problématique qui est de mieux sécuriser le monde maritime face aux cyberattaques, la chaire a su mettre en commun les capacités de plusieurs acteurs tels que l'école navale, IMT Atlantique, Thalès, Naval Group et l'ENSTA Bretagne.

Le monde maritime, en effet, a des spécificités qui lui sont propres. Les navires ont des durées d'exploitation qui s'étendent sur plusieurs décennies et cela a des conséquences sur la cybersécurité. Des systèmes qui étaient sécurisés hier ne le seront probablement plus demain. La chaire a donc trois objectifs qui découlent de cette problématique :

— permettre une meilleure détection des cyberattaques dans les systèmes navals.
— permettent d'établir une *cybersituation awareness* pour que les opérateurs des navires puissent être en capacité d'intervenir le cas échéant.
— permettre une meilleure conception des systèmes face aux cyberattaques et ainsi permettre une meilleure résilience de ces systèmes.

La thèse s'inscrit ici dans le dernier objectif, elle cherche en effet à rendre les systèmes plus résistants et résilients face aux cybermenaces, plus spécifiquement aux menaces sur l'opérateur humain qui conserve une part essentielle dans les systèmes navals.

1. http://www.senat.fr/questions/base/2021/qSEQ21021676G.html
2. https://cyber.forum.yale.edu/blog/2021/7/20/attributing-deaths-to-ransomware-attacks-on-hospitals-and-medical-care-facilities

1.2 Cadre des travaux de recherche

Le fait que la vulnérabilité humaine reste, dans nos systèmes, la vulnérabilité la plus importante est dû à plusieurs éléments. Le premier d'entre eux est la complexité grandissante des systèmes et leur évolution rapide. En effet, si nous prenons l'exemple du smartphone, celui-ci a connu une progression rapide dans les usages et dans les mœurs, à tel point qu'aujourd'hui, une grande partie de la population (environ 84%) l'utilisent plusieurs heures par jour alors qu'il y a une dizaine d'années, seule 30 % de la population en utilisait un [3]. Il en sera probablement de même pour l'usage des véhicules connectés qui, d'après les projections, pourrait devenir majoritaires dans le secteur [4]. Tout ceci illustre l'évolution rapide du numérique dans nos sociétés. Toutefois, toutes les tranches de la population ne sont pas exposées de la même manière à ce phénomène [3] et ne sont pas formées de la même manière.

Le deuxième élément est notre propension à être des individus imparfaits, incluant des biais dans nos raisonnements. Par exemple, le biais de confirmation ou le biais du survivant sont des biais très présents dans nos sociétés. Les élections américaines de 2016 ainsi que celle de 2020, pour ne citer que celles-ci en exemple, ont pu montrer la perméabilité des individus aux fausses nouvelles liées à l'usage du numérique. En effet, nous avons pu voir des polarisations d'individus sur des éléments informationnels complètement faux, ainsi que des raisonnements fallacieux au sein de cercles de confiance, sur certaines plates-formes ou réseaux sociaux [5]. Cette consommation massive d'information a bien montré les limites cognitives liées à l'absorption et à la vérification des informations. Plus nous sommes exposés à l'information moins nous la vérifions.

Le dernier élément est notre difficulté à maîtriser notre exposition au numérique. En effet, chacun d'entre nous va générer des traces de ses actions sur les réseaux qui pourront être exploitées sous forme de jumeaux numériques. Ainsi, des individus, des sociétés privées, des états vont pouvoir générer des conjectures à l'aide des informations liées à nos jumeaux numériques. Par exemple, Google fait partie des acteurs qui peuvent prédire des comportements chez leurs utilisateurs, comme par exemple, leurs probables futurs achats, et cela peut aller jusqu'à prédire des vagues épidémiques [6].

Ces différents exemples montrent la panoplie possible des risques qui pèsent sur un individu dans le monde numérique. Aucune solution ne pourra résoudre simplement ce problème, cela à cause de son côté multi-factoriel, mais aussi à cause de l'existence de plusieurs verrous scientifiques. Dans le cadre de cette thèse, nous avons choisi de chercher

à rendre les architectures des futurs SoSTS plus résistants à la vulnérabilité humaine. Pour ce faire, nous proposons une méthode d'ingénierie pour des architectes qui doivent concevoir ces systèmes. Cette thèse va donc se concentrer sur la détection et la simulation de la vulnérabilité humaine et son impact sur des architectures pour permettre de pallier, dès la conception, certains risques liés à la vulnérabilité humaine.

1.3 Questions de recherche

Dans le contexte de la vulnérabilité humaine, la thèse prend pour partie de se focaliser sur une méthodologie permettant d'estimer cette vulnérabilité dans un SoSTS, sa simulation, ainsi que la déduction de son impact sur des SoSTS. Trois questions de recherche ont été identifiées et seront développées dans cette thèse ainsi qu'une question de développement :

QR1 Un besoin méthodologique et de représentation : Comment peut-on représenter un SoSTS en prenant en compte la vulnérabilité humaine ? Quel type de méthodologie peut-on mettre en place pour utiliser ces représentations ? Ici, nos travaux doivent identifier quels processus peuvent permettre une représentation correcte de ces systèmes.

QR2 Un besoin de modélisation : Comment représenter la vulnérabilité humaine ? Comment représenter l'opérateur humain dans un SoSTS ? Quels sont les facteurs caractérisant la vulnérabilité d'un opérateur humain ayant une influence sur la sécurité ? Comment les évaluer, les utiliser ? Quelle métrique permet de quantifier cette vulnérabilité ?

QR3 Un besoin de simulation : Quelle méthode peut permettre de simuler la vulnérabilité humaine ? Quel impact cette vulnérabilité peut-elle générer sur un SoSTS ? Comment cette vulnérabilité peut-elle se propager dans un SoSTS ? Pour la simulation, il faudra lever le verrou consistant à trouver une méthode permettant d'intégrer les différents facteurs impliqués dans cette vulnérabilité.

QD1 Un besoin d'outils : Quel type d'outils peut permettre d'utiliser le langage de modélisation et les outils permettant de mettre en œuvre la simulation ? Comment développer cet outil, quelle technologie choisir pour permettre une mise en place simple en tant que démonstrateur, tout en permettant une utilisation par un utilisateur non expert ? La création de cet outil devra permettre l'application complète de notre approche (langage et méthode). Il devra, de plus, permettre la

manipulation de cas réels pour contribuer à la validation l'approche.

1.4 Démarche et contributions de la thèse

Pour répondre à la première question de recherche, nous nous sommes concentrés sur ce qui, pour un opérateur humain, participe à une vulnérabilité de cybersécurité : des faiblesses, des fragilités. Pour que l'estimation de la vulnérabilité humaine d'un opérateur soit la mieux contextualisée possible, nous nous sommes également intéressés à la modélisation des éléments de l'environnement pouvant influer sur un individu pendant une attaque cyber. La repose notre première contribution : une modélisation de l'humain dans les SoSTS en contexte cyber. Pour réaliser ce modèle, nous nous sommes appuyés notamment sur les sciences humaines et sociales, et en particulier, sur les travaux dans le domaine du *security management*. Pour compléter les éléments venant de la littérature, nous avons interrogé différents experts qui nous ont permis d'affiner notre méthode d'estimation de la vulnérabilité dans un SoSTS.

Pour la deuxième question de recherche, notre contribution est une méthodologie permettant à un architecte de modéliser un SoSTS à l'aide d'un langage de modélisation. Cette méthodologie va de paire avec le langage que nous avons créé : HoS-ML. Ce langage permet la représentation d'un SoSTS. Cette représentation permet d'inclure la modélisation de l'humaine et l'estimation de la vulnérabilité humaine. Ceci permet à l'architecte de visualiser l'impact d'une vulnérabilité humaine liée au cyber sur le système. La finalité de ces outils, méthodologie et langage, est d'identifier, à travers différents scénarios, (1) les vulnérabilités pouvant toucher le système et (2) les situations où un acteur peut être plus vulnérable qu'un autre vis-à-vis d'une menace. Pour permettre tout cela, nous nous sommes inspirés d'un langage existant qui permettait à l'origine de représenter les systèmes socio-techniques. De plus, pour permettre à l'architecte de représenter les opérateurs humains sans avoir une forte compétence dans le domaine, nous avons travaillé sur les facteurs composant la vulnérabilité humaine.

Pour ce qui est de la troisième question de recherche, nous avons choisi pour la simulation de nous baser sur une approche statistique. Plusieurs travaux existants dans la littérature nous permettent d'avoir accès aux probabilités d'influence que peut avoir un facteur sur un autre. À partir de ces travaux, nous avons créé un réseau bayésien qui constitue notre troisième contribution. Pour permettre une manipulation dynamique du langage et du réseau bayésien, nous avons créé un logiciel, basé sur Sirius, permettant à

un architecte de simuler de manière interactive différentes menaces sur l'architecture ; il s'agit de HoS-ML Editor.

Enfin, afin d'évaluer nos travaux, nous avons réalisé des expérimentations avec notre partenaire industriel et la marine nationale. Ces expérimentations sont des cas d'études dans lequel nous utilisons nos modèles, notre langage et la simulation. Ces cas d'études ont ensuite été évalués par des experts permettant une première comparaison avec les travaux qui ont été menés. Ces expérimentations et leurs résultats forment la quatrième contribution de cette thèse.

1.5 Plan

	QR1	QR2	QR3	QD1
Partie II	Chapitres d'état de l'art			
Chapitre 5	■		■	
Chapitre 6		■		■
Chapitre 7	Chapitre de validation			
Partie IV	Conclusion et perspective			

TABLE 1.1 – Matrice chapitres/questions de recherche

Pour répondre aux différentes questions de recherche, cette thèse suit la trame suivante :

Nous commençons, dans la partie II, au chapitre 2, par l'exposition du contexte autour de l'ingénierie des SoSTS. Puis le chapitre 3 aborde la sécurité par conception en présentant le contexte et l'état de l'art , et enfin, le chapitre 4 traite la question de la détection de la vulnérabilité humaine.

La partie III est consacrée à nos contributions. Dans le chapitre 5, nous détaillons nos apports liés à la modélisation du facteur humain et de sa vulnérabilité dans les SoSTS. Une fois ce modèle établi, nous l'utilisons pour mettre en place une estimation de la vulnérabilité humaine, ainsi qu'un calcul d'impact d'une telle vulnérabilité sur un SoSTS.

Dans le chapitre 6, nous décrivons le langage que nous avons défini pour représenter les modèles de vulnérabilité humaine et d'impact sur les SoSTS. Ce chapitre se concentre également sur la méthodologie qui permet l'usage de ce langage. Enfin, nous montrons l'outillage que nous avons réalisé pour faciliter l'utilisation de ces différents modèles.

Le chapitre 7 permet de mettre à l'épreuve les différents apports de la thèse en les confrontant à des cas d'études industriels. Pour mettre en place ces cas d'études indus-

triels, nous avons suivi une méthodologie qui permet d'apporter les premiers éléments de vérification.

La partie IV permet de dresser un bilan sur les apports de nos travaux ainsi que sur leurs actuelles limites et présente des perspectives possibles dans la suite de ces travaux.

Le tableau 1.1 récapitule, pour chaque chapitre, les questions de recherche qui sont traitées.

Deuxième partie

Contexte et état de l'art

Cette partie présente les différents éléments d'état de l'art à ce jour, et de contexte qui vont être utilisés dans cette thèse. Nous nous attacherons en particulier à définir ces éléments, analyser les moyens conceptuels permettant de les représenter dans un contexte d'ingénierie, et à identifier finalement les limitations des moyens conceptuels existants. Dans un premier temps, nous aborderons l'ingénierie des systèmes de systèmes socio-techniques. Nous présentons ensuite les principes de la sécurité par conception à travers le prisme qu'est l'évolution de la cybersécurité et des principales menaces qui en découlent. Enfin, le dernier chapitre concernera l'existant en termes de détection de la vulnérabilité humaine.

Chapitre 2

Ingénierie des systèmes de systèmes socio-techniques

Sommaire

Dans ce chapitre, nous présentons le domaine de recherche dans lequel se déroulent ce travail de thèse, à savoir l'ingénierie des systèmes de systèmes socio-techniques (SoSTS). Comme nous le verrons plus loin, cette dernière est un sous-champ disciplinaire de l'ingénierie système. Ce domaine prend en compte non pas un seul système complexe, mais une combinaison de systèmes complexes et humains mis en relation dans le but de fournir des fonctionnalités spécifiques.

Ce chapitre présente dans un premier temps la notion de SoSTS. Pour ce faire, nous partirons de la définition du terme de système pour ensuite passer par les systèmes de systèmes (SoS) et les systèmes socio-techniques (ST). Ensuite, nous exposerons brièvement l'ingénierie système, sa définition et son lien avec les SoSTS. Ces différentes définitions permettent d'une part d'exposer au lecteur les éléments de vocabulaire utilisés tout au long du présent rapport et d'autre part de donner les clés de compréhension de la problématique traitée.

2.1 Des systèmes aux systèmes de systèmes socio-techniques

Dans cette section nous nous intéresserons à la notion de SoSTS. Pour ce faire, nous devons d'abord exposer les éléments qui vont composer les SoSTS. Nous commençons par l'élément le plus élémentaire à savoir la notion de *système*, avant d'évoluer vers la notion de *système de systèmes* (SoS) en passant par celle de *système socio-techniques* (ST) pour enfin définir ce que sont les *systèmes de systèmes socio-techniques* (SoSTS).

2.1.1 Système

La notion de système peut être difficile à définir au vu de son usage dans le différents environnements et domaines. Cette multiplicité d'usage est notamment due à l'ancienneté du terme "système" qui, d'après le CNRTL [1], connaît sa première occurrence en 1552 dans la langue française avec comme définition "ensemble dont les parties sont coordonnées par une loi". Depuis, une multitude de définitions ont été proposées. Nous avons choisi de nous appuyer sur la norme ISO/IEC15288 [7] qui définit la notion de système comme suit :

> « Les systèmes sont considérés dans la présente norme internationale comme étant fabriqués par l'homme, créés et utilisés pour fournir des services dans des environnements définis au profit des utilisateurs et d'autres parties prenantes. Ces systèmes peuvent être configurés avec un ou plusieurs des éléments suivants : matériels, logiciels, humains, processus (par exemple, processus de révision), procédures (par exemple, instructions de l'opérateur), installations et entités naturelles (par exemple, eau, organismes, minéraux). Dans la pratique, on les considère comme des produits ou des services. La perception et la définition d'un système particulier, de son architecture et de ses éléments dépendent des intérêts et des responsabilités de l'observateur. Le système d'intérêt d'une personne peut être considéré comme un élément du système du système lui-même. Inversement, il peut être considéré comme faisant partie de l'environnement d'exploitation du système d'intérêt d'une autre personne. » (traduit de [7].)

Cette définition est complète, à large spectre et s'en trouve quelque peu difficile à assimiler. Aussi le *Systems Engineering Handbook* [8] est venue la synthétiser comme

1. https ://www.cnrtl.fr/etymologie/systeme

suit : « un système est une combinaison d'éléments en interaction organisés pour atteindre un ou plusieurs objectifs déclarés. » (traduit de [8].)

2.1.2 Systèmes de systèmes

Dans plusieurs domaines tels que la santé, le transport, la défense et bien d'autres, les opérations métiers reposent sur des systèmes séparés qui se sont mis en place de manière plus ou moins indépendante et qui collaborent afin de rendre possibles une ou plusieurs fonctionnalités qu'ils ne peuvent fournir de manière indépendante. Le bon fonctionnement des entités utilisant ces systèmes repose sur l'interaction de ces mêmes systèmes. Par exemple, lors d'un accident de la route, la répartition des ambulances va nécessiter la mise en œuvre de logiciels de gestion des parcs disponibles (qui peuvent être privés ou publiques), mais aussi du système d'information des différents centres de secours, des moyens de police, voire de gestion de la voirie. L'interaction de ces systèmes forme ce qu'on appelle un système de systèmes (abrégés SoS). L'intérêt des SoS est de permettre l'émergence de nouvelles propriétés, quelles soient fonctionnelles ou non-fonctionnelles, qu'aucun des systèmes utilisés ne peut fournir à lui seul.

Pour définir les SoS dans un premier temps, nous pouvons utiliser la définition de Maier de 1999 qui, aujourd'hui encore, est une des plus utilisées : « le terme SoS, couramment utilisé, suggère des assemblages de composants qui sont eux-mêmes significativement complexes pour qu'ils puissent être considérés comme des systèmes, et qui sont assemblés en un système plus vaste. » (traduit de [9].)

Cependant définir et qualifier un SoS est une problématique non triviale, et non récente. En effet, la notion de SoS remonte aux années 50. De ce fait, la définition a énormément évolué au fur et à mesure du temps et des technologies mises en œuvre.

La première définition de la notion de SoS est donnée en 1956 par [10]. Dans cette définition, on trouve déjà un certain nombre d'éléments toujours actuels pour qualifier le concept de SoS. Nielsen et al. [11] retracent toutes les définitions qui ont pu être données depuis la première définition de 1956 jusqu'à la définition proposée par l'INCOSE [2] en 2015 [12].

En se basant sur l'historique des définition, Nielsen extrait les 8 caractéristiques suivantes permettant de définir les SoS. Nous les résumons brièvement :

2. INCOSE : International Council on Systems Engineering. Société savante de référence dédiée à la promotion de l'ingénierie des systèmes. `https://www.incose.org/`

Autonomie Chaque système constituant a des fonctions qui lui sont propres et les règles qui s'appliquent sur ce système lui sont propres aussi et peuvent n'être définies que pour lui.

Indépendance Chaque système constituant peut opérer malgré le fait qu'il ne soit pas/plus connecté avec les autres systèmes constituants.

Distribution Les systèmes constituants communiquent entre eux pour permettre la répartition des tâches à réaliser ainsi que la potentielle réalisation de tâches en commun.

Reconfiguration dynamique C'est la capacité d'un SoS à créer des changements dans sa structure et sans que cela ne soit planifié par avance.

Évolutivité C'est la capacité d'un SoS à créer un changement, que ce soit sur ses fonctionnalités ou sur sa structure, pour permettre une évolution de manière planifiée.

Émergence Les actions cumulatives et les interactions entre les systèmes constituants d'un SoS donnent lieu à des comportements qui peuvent être attribués à l'ensemble du SoS.

Interdépendance Chaque système constituant est dépendant d'autres systèmes constituants au sein du SoS pour la réalisation d'un but commun.

Interopérabilité Les systèmes constituants peuvent communiquer entre eux malgré leurs potentielles hétérogènéités.

Un récapitulatif de l'usage de chacune de ces caractéristiques par les définitions existantes est présenté par la figure 2.1 tirée de [11].

2.1.3 Systèmes socio-techniques

Depuis l'émergence de la notion de système, une sous-catégorie spécifique est apparue pour identifier les systèmes dans lesquels l'être humain est partie prenante. Il s'agit des systèmes socio-technique (ST).

L'AFIS [3] propose de définir ce type de systèmes de la manière suivante [13] : « Un système socio-technique représente une organisation où une partie organisationnelle humaine va permettre la mise en œuvre d'un système technique permettant la réalisation d'une tâche. »

3. AFIS : Association Française d'Ingénierie Système - `https://www.afis.fr/`

Author(s)	Described in Section	Autonomy	Independence	Distribution	Evolution	Dynamic Reconfiguration	Emergence of Behaviour	Interdependence	Interoperability
Boulding [1956]	2.1	•	•					•	•
Ackoff [1971]	2.1	•						•	•
Eisner et al. [1991]	2.5			•			•	•	•
Noam [1994]	2.4	•	•	•			•		•
Shenhar et al. [1994]	2.3			•			•		•
Manthorpe [1996]	2.4				•		•		•
Maier [1996]	2.2	•	•	•	•		•		
Kotov [1997]	2.4		•	•					
Lukasik [1998]	2.5			•	•		•		•
Krygiel [1999]	2.7			•	•		•		•
Roe [1999]	2.5			•	•		•		•
Cook et al. [1999]	2.7	•	•	•	•		•		•
Pei [2000]	2.4			•		•			•
Carlock and Fenton [2001]	2.4		•		•		•	•	•
Sage and Cuppan [2001]	2.7	•	•	•	•				•
Chen and Clothier [2003]	2.5	•	•		•	•	•	•	•
Keating et al. [2003]	2.5	•	•			•	•	•	
Bar-Yam et al. [2004]	2.7		•		•		•	•	•
Crossley [2004]	2.5		•		•	•			•
DeLaurentis and Crossley [2005]	2.3	•	•	•	•	•	•	•	
Abbott [2006]	2.2				•	•			
Boardman and Sauser [2006]	2.2	•		•	•	•	•		•
Cocks [2006]	2.4	•		•	•		•		
Boehm [2006]	2.4				•		•		
Fisher [2006]	2.7	•	•	•	•		•	•	
Sharawi et al. [2006]	2.7	•	•	•	•	•	•		•
DoD SE Guide for SoS [2008]	2.6	•	•	•	•	•	•	•	•
Karcanias and Hessami [2010]	2.3	•	•	•	•		•	•	
INCOSE [2015]	2.6	•	•	•	•	•	•	•	•
Count	–	16	17	19	21	10	22	12	20

FIGURE 2.1 – Historique des définitions des systèmes de systèmes, extrait de [11]

L'être humain, en tant qu'entité d'un système, prend une place importante dans la modélisation des ST. En effet, la réalisation des fonctions du système est le résultat des actions menées par les acteurs humains combinées aux actions des différents systèmes techniques . Ceci est dû au fait que l'appropriation et l'adaptation du système technique par l'opérateur humain, dans l'optique de la réalisation d'une mission, est nécessaire. De plus, cela peut permettre d'apporter de la flexibilité au ST, car l'opérateur humain va généralement apporter, par ses capacités d'analyse, une plus grande souplesse et de meilleures adaptations et réactions du ST face à son environnement.

Cependant, les ST apportent aussi de nouvelles problématiques. L'être humain permet certes une adaptation du ST face à son environnement, mais va potentiellement être une nouvelle source de problème. En effet, dès les années 60, de multiples incidents mettent en lumière le fait que l'opérateur humain peut être un élément nuisible au système, tout en étant un élément interne au système [14]. Il va également y apparaitre des problèmes liés à l'ergonomie pour l'opérateur s'appropriant le système. Ces multiples nouveaux problèmes émergeants conduisent à mettre en place de nouvelles méthodes, aussi bien dans l'ergonomie, pour permettre à l'opérateur de mieux réaliser sa tâche, que dans le domaine de la sécurité du ST. Ce dernier domaine prend le nom de résilience ou de robustesse selon les articles de la littérature. Les deux termes étant reconnus comme synonymes dans ce cas [15]. Dans ce document, nous utiliserons le terme résilience pour les ST.

2.1.4 Systèmes de systèmes socio-techniques

Le concept de systèmes de systèmes socio-techniques (SoSTS) représente la fusion entre les systèmes de systèmes (SoS) et les systèmes socio-techniques (ST). En effet, dans les systèmes actuels, que ce soient les systèmes de santé ou les systèmes militaires par exemple, la complexité de ce type de ST lui confère le status de SoS. Ainsi, parce que certains éléments composant ce type de SoS sont des ST, il devient nécessaire d'intégrer les propriétés des ST dans les SoS pour mieux s'adapter à l'environnement global et à l'ensemble des émergences comportementales.

On peut définir un SoSTS de la manière suivante [16] : « un système de systèmes socio-technique est une collection de systèmes à la fois techniques et socio-techniques qui mettent en commun leurs capacités à faire partie d'un système plus complexe, tout en conservant leur autonomie. »

Cette première définition met en évidence l'autonomie comme pour les SoS classiques. En effet, on peut légitimement reprendre chacune des caractéristiques définissant un SoS

et présentées en amont. Il faut ajouter à cela les caractéristiques des ST et notamment leur capacité d'adaptation.

2.2 Ingénierie système

Être capable de construire des systèmes complexes a toujours été un objectif pour l'humain. De tout temps, l'homme a cherché à développer des systèmes afin de l'aider dans ses tâches, d'abord en architecture, puis dans la construction de machines. Maîtriser les techniques et outils permettant de construire ces machines a longtemps été l'apanage de maîtres, formés par apprentissage selon des savoirs empiriques. Ainsi est née ce que l'on pourrait appeler l'ingénierie système. La formalisation de ce domaine est cependant apparue avec la mécanisation et la complexification des systèmes, principalement liés à la révolution technologique de l'après deuxième guerre mondiale [8]. Aujourd'hui, ce phénomène s'accentue toujours. Les concepts portés par l'ingénierie système sont évidemment toujours d'actualité, mais nécessitent de les faire évoluer au fur et à mesure que sont identifiés de nouvelles connaissances et de nouveaux besoins. Pour illustrer cette complexification, nous pouvons énoncer le fait qu'il existe aujourd'hui au moins 35 standards pour définir l'application de l'ingénierie système sur des domaines différents [17]. Les domaines d'application vont du spatial à la défense ainsi qu'à la conception de systèmes industriels tels que des navires.

2.2.1 Définition de l'ingénierie système

L'ingénierie système est un domaine qui s'est formalisé il y a plus de 70 ans. Aussi beaucoup de définitions ont été données et n'en choisir qu'une revient à choisir celle qui fait le plus consensus ou sera la plus complète. Nous prenons comme référence la définition portée par l'INCOSE qui semble faire consensus de par sa dimension normative. L'ingénierie système y est décrite de la manière suivante :

> « L'ingénierie système est une approche interdisciplinaire et un moyen pour permettre la réalisation de systèmes performants. Elle se concentre sur la définition des besoins du client, des fonctionnalités requises au début du cycle de développement, la documentation des exigences, puis la conception du système pour finir par la validation du tout en considérant le problème dans son ensemble. L'ingénierie système tient compte à la fois des besoins commerciaux et des besoins techniques des clients dans le but de fournir un produit de qualité qui répond aux besoins des utilisateurs. » (Traduction de [18].)

Cependant, un élément, qui nous semble important, est absent dans cette définition. Il s'agit du fait que l'ingénierie système n'est pas simplement un domaine scientifique, mais aussi comme, le fait remarquer la NASA dans sa définition, un art :

> « L'ingénierie système c'est l'art et la science de développer un système exploitable et capable de répondre aux exigences dans le cadre de contraintes souvent opposées. » (Traduit de [19].)

Effectivement, il faut savoir composer avec des domaines qui peuvent être souvent très différents et qui vont avoir des contraintes qui peuvent se contredire. Aussi l'usage de l'ingénieur système sur un système complexe passe souvent par une adaptation face aux spécificités de celui-ci, qui nécessite un certain niveau de créativité.

La complexité des propriétés à prendre en compte dans la construction des systèmes continue d'ailleurs de croître. Dans la présente thèse, nous nous intéressons particulièrement à un type : la prise en compte des propriétés humaines pouvant avoir un impact sur le système. En effet, une meilleure connaissance des propriétés humaines et de leurs relations avec les systèmes complexes ont permis de développer de nouvelles techniques et méthodes améliorant la conception des systèmes. Il a fallut intégrer ces nouveaux concepts, faisant ainsi émerger de nouveaux domaines tels que les SoSTS, discutés précédemment. Avec cette émergence, est apparu le besoin d'adapter l'ingénierie système de manière à les prendre en compte.

2.2.2 Cycle de conception en ingénierie système

De nombreuses méthodologies d'ingénierie système ont émergé au fur-et-à-mesure des besoins [20] de conception et d'usage. Afin d'illustrer les grands principes de l'IS, nous allons prendre l'exemple d'une méthodologie en particulier et montrer son adaptation face à des besoins spécifiques. La méthodologie du cycle en V [21] tire son nom de la forme que l'on donne aux différentes étapes composant la conception du système, car celles-ci vont former un V où chaque étape va répondre à une autre étape (voir Figure 2.2). Le chemin de gauche représente les phases de conception et le chemin de droite représente les phases de vérification. À chaque phase de conception correspond une phase de vérification.

Cependant l'usage de cette méthodologie s'est avéré plus complexe lorsqu'il a fallu changer de paradigme en passant de l'ingénierie système à l'ingénierie des systèmes de systèmes. En effet, l'ingénierie système est centrée sur la définition d'un système alors que

4. source : Wikimedia Commons, image non modifiée, auteur : Cth027 , licence : CC BY-SA 4.0.

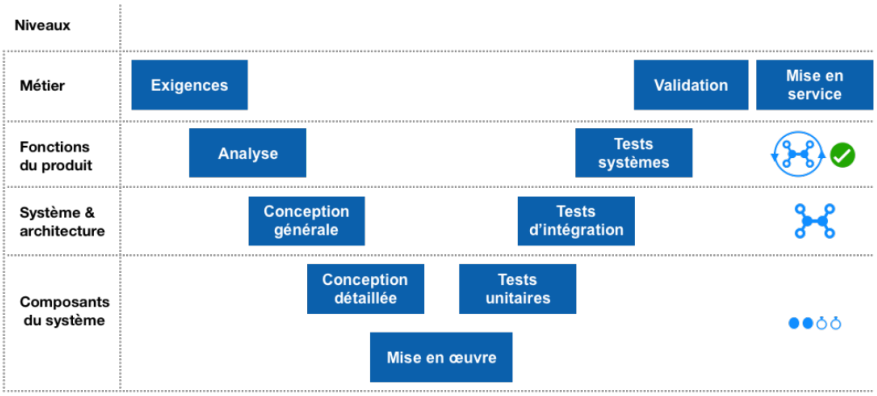


FIGURE 2.2 – Illustration d'un cycle en V en ingénierie système [4]

FIGURE 2.3 – Illustration du cycle en double V [22]

l'ingénierie des systèmes de systèmes se focalise plutôt sur les interactions entre systèmes. Une adaptation de l'approche en V a été nécessaire. Elle a pris la forme du cycle dit en double V [22], qui permet de mieux définir à la fois le système de systèmes et les systèmes qui le composent. La Figure 2.3 montre l'adaptation qui a été faite du cycle en V, notamment en répétant avec de plus petites itérations le cycle en V sur les systèmes composant le SoS.

2.2.3 De l'ingénierie système à l'ingénierie des SoSTS

Le passage de l'ingénierie système à une ingénierie manipulant des concepts tels que ceux portés par les systèmes de systèmes socio-techniques est notamment due à la mo-

FIGURE 2.4 – Illustrations des différents éléments à prendre en compte lors de la conception d'un SoSTS [23]

dernisation des différents types de systèmes et à l'émergence de leur coopération aussi bien entre eux qu'entre les différents utilisateurs de ces mêmes systèmes. Dans sa vision 2035 l'INCOSE [1] montre que l'abondance de nouveaux systèmes coopérants et le besoin d'interopérabilité entre eux et les individus qui les utilisent ou les servent a posé de nouveaux défis en termes de conception de ces systèmes. Pour ce faire, le concept de système de systèmes socio-techniques a été apporté au début des années 2010 [16] pour permettre de répondre à un certain nombre de problèmes dans la conception des architectures modernes.

Un cas typique est aujourd'hui celui des réseaux électriques intelligents ou *smart grid*. Ces réseaux cherchent à répondre aux contraintes environnementales grandissantes et aux tensions énergétiques que peuvent connaître les différents acteurs mondiaux. Les réseaux intelligents sont typiquement des structures qui deviennent essentielles et qui vont avoir besoin d'une conception de plus en plus poussée [23]. Cette complexité dans la conception vient notamment de l'intégration d'éléments dans ces réseaux tels que sont les compteurs intelligents. Ces compteurs permettent la prise en compte en temps réel de différents types de consommation des différents équipements. Ce nouveau type d'équipement permet une grande finesse sur la gestion de l'approvisionnement, mais au détriment de véritables contraintes sur la modélisation et la complexité autour de ces réseaux. À ces éléments vient s'ajouter la notion de facteurs humains. Ce facteur humain s'inscrit dans la modélisation

de l'utilisateur final. Par exemple, dans ses habitudes de consommation qui demandent de la part du réseau intelligent une réactivité voire une interaction. Le facteur humain peut aussi venir s'inscrire dans la modélisation de la prise en compte de l'erreur humaine lors de l'exploitation de ces grilles intelligentes pour permettre une meilleure résilience du SoSTS. La figure 2.4 montre ainsi les différents éléments constituant une telle conception. On peut en effet y voir que de nombreuses nouvelles préoccupations sont à prendre en compte par les ingénieurs, venant s'ajouter aux processus de modélisation du système de systèmes. Ainsi, en plus des systèmes techniques, dans ce cas précis notamment, il faudra aussi considérer le facteur humain, la sûreté et la sécurité, les opérations de maintenance, etc.

En effet, prenons l'exemple de l'étude, présenté dans [24], où les auteurs avaient le besoin de modéliser aussi bien les différents secteurs techniques faisant partie de l'architecture que les interactions entre les humains et les machines. Ils ont également été amené à modéliser, dans une certaine mesure, les éléments environnementaux pouvant avoir des influences sur l'architecture, tels que la météo. Ce sont ces intégrations qui rendent nécessaire le passage aux SoSTS.

Les moyens de conception soutenant la modélisation des SoS, ST ou SoSTS visent, donc, à permettre la modélisation de l'interaction entre ces systèmes techniques ou humains, mais également la prise en compte des comportements émergents liés à la mise en relation de ces systèmes. Concernant les problématiques de sécurité, il est possible de les prendre en compte sous l'angle de la maîtrise de comportements émergents non désirés. En effet, l'ingénieur cherchera ainsi à mieux prévoir les effets de bord de certains systèmes sur d'autres.

2.3 En résumé

Les systèmes actuels sont de plus en plus complexes. Ils interagissent souvent entre eux de manière plus ou moins naturelle de manière à fournir des fonctionnalités ou des propriétés que seuls, les systèmes ne peuvent fournir. De nouveaux paradigmes ont été nécessaires pour les décrire. Dans ces nouveaux paradigmes, la prise en compte de l'humain a été rendue nécessaire à cause de la nature, difficilement prédictive, de l'humain. Nous nous intéressons dans, le cadre de cette thèse aux systèmes de systèmes socio-techniques. Généralement, on parle d'ingénierie dans un domaine pour identifier les savoirs industriels, les techniques et les méthodes propres à un domaine. Malgré le fait que le concept de

SoSTS soit relativement récent, et que la définition même du concept ne soit pas encore consensuelle, il devient évident que les différentes notions qu'il apporte permettront de modéliser des systèmes de manière plus complète et d'y associer l'humain de manière plus importante. Cependant, le fait que le SoSTS se détache des concepts existants montre la nécessité d'avoir une ingénierie système qui lui soit propre. Cela est d'autant plus vrai lorsque l'on aborde la potentielle vulnérabilité des SoSTS du point de vue sécuritaire. Associer le facteur humain aux systèmes techniques revient à réunir deux mondes qui ont déjà des vulnérabilités intrinsèques et dont l'association peut conduire à l'émergence de nouvelles formes de vulnérabilités. Le prochain chapitre traite donc de la cybersécurité, de ses enjeux et de sa prise en charge dans le cadre de l'ingénierie des systèmes de systèmes socio-techniques.

Chapitre 3

Sécurité par conception

Sommaire

Ce chapitre se concentre sur la sécurité par conception. Pour exposer les enjeux liés à la sécurité par conception, nous commençons par présenter le concept de cybersécurité. Nous poursuivons notre propos en présentant la manière de sécuriser les systèmes, qu'ils soient déjà existants ou lors de leur conception. Nous discutons ensuite des méthodologies existantes de sécurité par construction autour des SoS et des ST. Nous terminons enfin sur la sécurité par construction appliquée aux SoSTS.

3.1 La cybersécurité

La cybersécurité est un domaine qui est aujourd'hui en pleine effervescence, cela notamment depuis la recrudescence des attaques perpetrées durant la période de la crise sanitaire [25]. Cette période a su rappeler, s'il le fallait, que notre usage de plus en plus grandissant des systèmes connectés impose de lier cet usage avec une forte cybersécurité. En effet, des secteurs que l'on croyait relativement préservés en raison de leurs fonctions essentielles, comme les hôpitaux, ont connu des cyberattaques venant directement nuire à leurs intérêts vitaux. Afin de bien prendre en compte cette propriété essentielle qu'est la cybersécurité, il convient de bien la définir.

3.1.1 Définition

La cybersécurité est définie par ANSSI [1] comme étant :

> « un état recherché pour un système d'information lui permettant de résister à des événements issus du cyberespace susceptibles de compromettre la disponibilité, l'intégrité ou la confidentialité des données stockées, traitées ou transmises et des services connexes que ces systèmes offrent ou qu'ils rendent accessibles. La cybersécurité fait appel à des techniques de sécurité des systèmes d'information et s'appuie sur la lutte contre la cybercriminalité et sur la mise en place d'une cyberdéfense. » [2]

Cette définition met notamment en lumière les propriétés qui sont constitutives de la cybersécurité : la disponibilité, la confidentialité et l'intégrité. Ces propriétés ne sont pas propres à la définition donnée par ANSSI, qui rappelons le est un organisme français. Si nous prenons une définition plus internationalement reconnue, celle par exemple de l'Union Internationale des Télécommunications(UIT) [26], ces trois propriétés apparaissent de nouveau et avec une place centrale.

Pour bien comprendre la définition de la cybersécurité, nous allons donc nous intéresser à ces propriétés ainsi qu'au vocabulaire attenant.

3.1.2 Vocabulaire lié à la cybersécurité

La cybersécurité a, comme bien des domaines, son propre vocabulaire. Nous allons définir les propriétés ainsi que le vocabulaire qui sont utilisés dans cette thèse. Pour définir ces termes, nous nous appuyons sur des sources telles que l'ANSSI qui fait office de référence nationale ainsi que sur des normes internationales telles que celles définies par l'ISO [3] (ISO-27000 [27]) et le NIST [4] [28], références en la matière.

Ainsi, nous retenons pour les trois principales propriétés de sécurité les définitions suivantes (traduit de [27]) :

Confidentialité Propriété d'une information qui n'est ni disponible, ni divulguée aux personnes, entités ou processus non autorisés.

Intégrité Garantie que le système et l'information traitée ne sont modifiés que par une action volontaire et légitime.

1. Agence Nationale de la Sécurité des Systèmes d'Information - `https://www.ssi.gouv.fr/`
2. https ://www.ssi.gouv.fr/entreprise/glossaire/c/
3. ISO : International Organization for Standardization
4. National Institute of Standards and Technology

Disponibilité Propriété que le système soit accessible et utilisable sur demande par une entité autorisée.

Le vocabulaire que nous choisissons ici de définir est celui qui va être utilisé au cours de cette thèse. Nous utilisons plusieurs éléments venant du monde de la cybersécurité qui peuvent avoir des sens différents dans d'autres domaines :

Menace : Cause potentielle d'un incident indésirable, qui peut entraîner un préjudice pour un système ou une organisation (Traduit de ISO [27]).

Vulnérabilité : Faiblesse d'un système, des procédures de sécurité du système, des contrôles internes ou de la mise en œuvre qui pourrait être exploitée ou déclenchée par une menace (Traduit de NIST [28]).

Attaque : Tentative de détruire, d'exposer, d'altérer, de désactiver, de voler ou d'obtenir un accès non autorisé à un bien ou de faire un usage non autorisé d'un bien (Traduit de ISO [27]).

Politique de sécurité : Un ensemble de règles qui régissent tous les aspects du comportement d'un système et des éléments du système (Traduit de NIST [28]).

Pour permettre de visualiser plus facilement l'usage des différentes définitions présentes ci-dessus. Voici un exemple venant illustrer chacune de ces définitions.

Soit le système d'information d'une entreprise, risquant de subir une attaque de vol de mot de passe. Un groupe de hacker nommé REKCAH représente la menace. Il utilise une vulnérabilité sur le gestionnaire de mot de passe et rend ainsi caduque la politique de sécurité demandant un tunnel sécurisé lorsque l'utilisateur entre son mot de passe.

3.2 Ingénierie et cybersécurité

Pour traiter la cybersécurité des systèmes, il peut y avoir deux situations. Dans la première, le système est déjà existant et dans ce cas, le cybersécuriser passe par une évaluation de l'existant et de sa sécurisation. C'est une sécurisation *a posteriori* de la création du système. Dans la seconde situation, le système n'existe pas encore. Ainsi, sa sécurisation peut être faite *a priori*, c'est-à-dire en mettant en place la politique de sécurité dès la phase de conception. La thèse présentée ici se concentre sur la deuxième situation : nous cherchons à modéliser les SoSTS de manière sécurisée dès leur conception. Aussi, nous allons, dans un premier temps, présenter les principes de conception des systèmes via des

méthodes d'ingénierie système. Ensuite, nous présenterons les méthodologies existantes permettant d'intégrer à cette dernière des principes de sécurité dès la phase de conception.

3.2.1 Sécurité par construction

Faire de la sécurité dès la conception, ce n'est pas juste mettre en place à un moment donné des éléments pour couvrir certains risques. Il faut aussi prendre en compte l'intégralité de la vie du système. Pour ce faire, le "Security Development Life Cycles" (SDLC) est une approche portée notamment par le NIST [29]. Elle indique que même si un système a été conçu comme sécurisé, sans maintenance intégrant la sécurité, il peut devenir non sécurisé. La sécurité dès la conception doit donc prendre un maximum d'éléments en compte pour permettre au systèmes une bonne adaptation au fur-et-à-mesure du temps.

Afin de répondre à ce besoin de sécurité tout au long du cycle de vie du système, il existe plusieurs méthodologies applicables en fonction des besoins de sécurité ainsi que des systèmes que l'on souhaite modéliser. Nous en présentons quelques-unes ci-après. Les approches existantes concernent, majoritairement, les cas des SoS et ST. En effet, à notre connaissance, il n'existe pas encore d'approche proposée pour couvrir le cas des SoSTS.

3.2.1.1 Sécurité par construction des SoS

Dans la suite, nous nous intéressons à plusieurs approches de modélisation de SoS prenant en compte les propriétés de sécurité. La liste des approches présentées n'est pas exhaustive, le choix s'est porté sur des approches qui nous semblent représentatives et qui chacune traitent certains aspects de la sécurité. Par exemple, la capacité à représenter le chemin d'un attaquant ou encore celle de représenter des notions de contraintes de sécurité à certains endroits du SoS.

Un profil SysML spécifique Dans [30], les auteurs proposent une extension du langage SysML, utilisant le mécanisme des profils, qu'ils développent selon un processus d'ingénierie dirigée par les modèles (MDE). Ils l'utilisent pour modéliser des scénarios pour les smart grid. Les auteurs proposent une intégration de la sécurité dans les SoS pour certains concepts de sécurité auxiliaire au SoS tels que le chiffrement et l'accès aux différents moyens de contrôle du SoS. Ils permettent également la possibilité de détecter des incidents de sécurité liés à des vulnérabilités. Les auteurs mentionnent également la possibilité de personnaliser leur outil pour générer du code, en vue d'une analyse ultérieure en utilisant des techniques d'analyse des

risques. Aucune de leurs suggestions discutées ne sont cependant mises en œuvre. Ce travail est encore à un stade précoce, seule l'extension SysML et son éditeur ont été implémentés.

IoTSAT Dans [31], les auteurs proposent un framework pour analyser la sécurité de l'internet des objets (IoT). Ils analysent la configuration du réseau par rapport à différentes menaces liées aux IoT. D'après les auteurs, il s'agit de la première tentative de modélisation et d'analyse du comportement général d'une architecture IoT et de sa robustesse face aux menaces. Cette étude peut aider à analyser la robustesse d'un réseau IoT contre des scénarios d'attaques spécifiques. Cependant, la modélisation des IoT se fait à l'aide de graphes, sans tenir compte des caractéristiques spécifiques des SoS, tels que l'indépendance et le comportement émergeant. Cette approche vise une représentation spécifique d'une application IoT au niveau de sa couche réseau, et non pas à la vision SoS.

SoSSecML Dans [32], l'auteur a mis en lumière d'autres méthodes liées à une conception sécurisée. L'approche proposée doit permettre la prise en compte d'éléments de sécurité ou au moins d'en permettre l'intégration. Pour ce faire, l'auteur modifie le langage SysML de manière à lui ajouter différents éléments permettant la modélisation de la sécurité. Notamment, il utilise, dans la modélisation de la vulnérabilité, des pré-conditions et des post-conditions pour la qualifier. Ce langage intègre aussi la notion d'organisation et des mesures préventives pouvant potentiellement empêcher les vulnérabilités d'être exploitées par l'attaquant. Tous ces éléments permettent notamment la représentation des attaques en cascade qui sont souvent un élément important à prendre en compte dans les SoS.

Ces différentes approches de modélisation pourront être complétées ensuite par une batterie de tests sur le SoS, mais aussi par des outils de simulation. Ainsi, la possibilité de simuler le comportement du SoS va permettre de mieux s'adapter aux scénarios d'attaque. De plus, les simulations peuvent permettre d'anticiper de potentiels comportements émergeants. En effet, à travers les différents scénarios, des comportements non prévus peuvent faire leur apparition. La simulation permettra donc le cas échéant de corriger ces comportements ou d'en limiter l'impact sur le SoS. Cela est mis en pratique dans [33] à travers des scénarios pouvant être appliqués par un outil de simulation sur le SoS.

3.2.1.2 Sécurité par construction des ST

L'ingénierie des SoS considère généralement les systèmes constituants comme des systèmes techniques ou des organisations. Le concept d'opérateur humain n'est que rarement présent et la plupart du temps, il n'est pris en compte que dans des cas d'usage des systèmes constituants et non pas comme un élément à prendre en compte tel quel dans le SoS. Aussi la modélisation de l'opérateur humain ainsi que l'évaluation de sa vulnérabilité ne sont pas pris en compte dans les SoS. Nous allons maintenant aborder les systèmes socio-techniques qui pallient ce manque en focalisant explicitement sur l'humain lors des phases de conception.

L'idée d'inclure conceptuellement la prise en compte de l'opérateur humain dans le fonctionnement des systèmes, en tant qu'élément à sécuriser, est de plus en plus présente [34]. Le but est d'essayer de le rendre moins vulnérable à une attaque. À notre connaissance, peu d'approches conceptuelles permettent à ce jour de modéliser la vulnérabilité d'un ST vis-à-vis des attaques via un opérateur humain. Nous avons cependant identifié deux approches différentes : La première concerne les systèmes de grande taille et nécessite une expertise forte. La seconde, concerne les systèmes de plus petite taille et nécessite une expertise plus faible :

STS-ML [35] Ce langage a été créé au début des années 2010 afin de permettre la création de systèmes socio-techniques sécurisés et résilients. Il concerne plus exactement l'étape d'ingénierie des exigences. Le langage permet de conceptualiser un ST à un haut niveau d'abstraction et d'en tirer un certain nombre d'exigences pour permettre au mieux sa sécurisation. Pour ce faire, le langage compte trois vues :

- *Social view* : cette vue permet la représentation de l'architecture du ST à travers les concepts de rôles et d'acteurs. Un rôle représente un opérateur humain idéal (celui imaginé par l'architecte du système) et un acteur représente l'opérateur humain occupant le poste idéal, mais avec parfois des caractéristiques différentes. Cette vue fait également référence aux différents objectifs et documents associés aux rôles et acteurs. Les objectifs permettent de définir les missions attribuées aux rôles et aux acteurs, les documents eux permettent de définir les données en possession des rôles ou acteurs. Cette vue permet aussi de voir les flux entre les différentes parties prenantes du système qu'il soit acteur ou rôle.Enfin, la vue permet de positionner des menaces.

Information view : cette vue permet la représentation des différents documents qui sont en possession des différents rôles et acteurs. Elle permet aussi de définir quels documents font partie d'autre document. Cela permet de visualiser la composition des différentes documents en possession des différents rôles et acteurs.

Authorization view : cette vue permet la représentation des différents documents et objectifs qui sont en transmission ou en délégation vers un autre rôle ou acteur. À travers ce flux d'un acteur vers un autre acteur, il est possible de définir des autorisations (écriture, lecture ou modification) pour chaque acteur sur ses objectifs et ses documents.

Les trois vues proposées dans ce langage permettent notamment d'estimer l'impact potentiel d'une menace sur le système en modifiant les caractéristiques d'un acteur ou d'un rôle. Cette estimation se fait en fonction des éléments présents dans les trois vues et va permettre une analyse à la fois visuelle (sur les modèles) et détaillée dans un rapport généré. Ce dernier établit un certain nombre d'exigences à remplir pour rendre le système à la fois résiliant et sécuritaire vis-à-vis de l'architecture définie.

CJML & SSM [36] : Ce langage a été créé à partir de deux autres éléments qui sont Customer Journey Modelling Language (CJML), qui est un langage visuel pour la modélisation et la visualisation des processus de travail, en termes de parcours utilisateur, et System Security Modeller (SSM) qui est un outil d'architecture et d'analyse des risques pour les systèmes socio-techniques. Ce langage a pour but de permettre aux petites et moyennes entreprises (PME) de définir leurs processus et leurs systèmes socio-techniques de manière simple. Ce langage cherche notamment à introduire le facteur humain, dans la phase de conception, à travers la notion de processus. Ces processus aident à mettre en évidence les potentielles menaces qui peuvent affecter un individu dans son travail. Le tout est mené en reliant ces modèles à une architecture de systèmes socio-techniques et en appliquant dessus une analyse de risque.

Les approches d'ingénierie visant à spécifier et modéliser un ST, bien qu'ils permettent la définition d'un opérateur humain, ne permettent pas réellement de définir la vulnérabilité humaine ni de prendre en compte la notion de SoS. Ces deux points sont importants, car d'une part, l'émergence, qui est une importante source de vulnérabilité, est une conséquence de l'inter-connexion entre les systèmes composant le SoS et, d'autre part, l'humain

en tant que composant du SoS est une autre source importante de vulnérabilité. Il est donc plus judicieux de prendre en charge les problème liés à la cybersécurité à l'échelle des SoSTS.

3.3 En résumé

Pour sécuriser les SoSTS par construction, il est important de prendre en compte les spécificités de ces derniers. Les SoSTS sont une agrégation des concepts de SoS et de ST. Ainsi, les systèmes constituants peuvent être soit techniques, soit humains. Or, considérer l'opérateur humain en tant que sous-système, pouvant lui-même être une source de vulnérabilité, n'est que peu pris en compte par les approches proposées jusqu'à présent. Il est donc important de prendre en compte les facteurs humains pour modéliser l'opérateur dans un SoSTS. Il devient alors possible d'étudier l'impact qu'il peut avoir sur le SoSTS et son environnement.

Chapitre 4

Détection de la vulnérabilité humaine

Sommaire

Dans ce chapitre nous traitons de la détection et de l'estimation de la vulnérabilité humaine. Pour ce faire, nous nous interrogerons dans un premier temps sur comment peut-on modéliser l'humain avec comme problématique la cybersécurité. Nous définirons ce qu'est dans ce cadre un facteur humain, puis exposerons les différents outils conceptuels existants autour de ce concept. Dans un second temps , nous aborderons l'estimation de la vulnérabilité humaine à travers deux approches différentes qui sont l'approche par expertise et l'approche outillée.

4.1 Modéliser l'humain

Modéliser l'humain dans sa complexité s'avère trop compliqué. Les travaux visant à représenter l'humain d'un point de vue conceptuel se focalisent donc sur un ensemble de propriétés significatives pour l'aspect que l'on cherche ici à modéliser. On parle alors de facteur humain. Si l'on restreint le champ d'étude autour des relations entre un humain et un système, là encore le domaine des possibles reste bien trop vaste. Ainsi, les travaux

menés sur ce sujet sont souvent très spécialisés. On peut observer par exemple les études qui sont faites sur les IHM (Interface Humain Machine) et leur facilité d'usages qui vont utiliser abondamment les travaux issus des sciences sociales afin de concevoir des IHM plus adaptées à l'humain.

Les travaux menés dans cette thèse visent à mieux prendre en compte la vulnérabilité humaine dans un contexte cyber. En effet, le facteur humain est aujourd'hui responsable de la majorité des attaques qui réussissent sur des systèmes. Dans ce chapitre, nous allons dans un premier temps nous concentrer sur les modélisations existantes du facteur humain cherchant à réduire la surface d'attaque des individus. Dans un deuxième temps, nous allons explorer les approches visant à estimer quantitativement la vulnérabilité humaine.

4.1.1 Définition

La notion de facteur humain est une notion qui vient à l'origine des études scientifiques portant sur l'ergonomie. Cette dernière est un domaine scientifique visant à développer la meilleure interaction possible entre l'homme et la machine, cela afin d'en améliorer l'efficacité. Les travaux sur le facteur humain ont été longtemps menés de manière plus ou moins formelle. On note une première ébauche de formalisation après la Deuxième Guerre mondiale [37].

La définition du concept de facteur humain a évolué au fur et à mesure de l'évolution des systèmes ainsi que de la compréhension de l'interaction entre l'homme et les systèmes. Nous retenons la définition suivante issue de [38], [39] : « *On peut définir le facteur humain comme faisant référence aux différents facteurs qu'ils soient organisationnels, environnementaux, professionnelles ainsi qu'aux caractéristiques humaines composant des individus et leurs limites qui peuvent influencer le comportement humain et impacter les systèmes et organisations l'entourant.* »

Le facteur humain a été perçu dans le monde de la cybersécurité comme un enjeu important. Il a fait l'objet de plusieurs publications a partir des années 2000 : [40], [41]. Différentes méthodes ont émergé pour essayer de canaliser le facteur humain et de réduire sa vulnérabilité. [42] propose une revue de littérature sur la question de la sécurité informatique et du facteur humain. Il identifie cinq points permettant de limiter la vulnérabilité humaine dans le monde de la sécurité informatique :

Politique de sécurité de l'information Mettre en place une politique de sécurité bien définie. Cette nécessité peut venir d'une obligation réglementaire à l'origine,

voire législative. Elle va conditionner un certain nombre d'éléments comme des certifications ou un certain nombre d'audits. Cette politique de sécurité va influencer fortement le comportement des utilisateurs. En effet, ils vont se voir imposer un certain nombre de règles qui vont limiter favorablement la vulnérabilité humaine.

Dissuasion et incitation Lorsque des politiques de sécurité sont mises en place dans une organisation, il est recommandé de définir des moyens de sanction permettant de faire respecter ces politiques. Il a été observé que lorsque des sanctions vis-à-vis d'un manquement au respect de la politique de sécurité sont sévères, l'organisation développe une meilleure culture de la sécurité.

Attitudes et implication Afin de disposer d'une bonne culture de sécurité dans une organisation, il a été montré qu'il fallait aider les employés à adopter des attitudes positives vis-à-vis de la culture de la sécurité en général. Il est nécessaire également de les impliquer dans les attendus de sécurité définis par l'organisation.

Formation et sensibilisation Afin d'aider à la mise en place d'une bonne culture de sécurité dans une organisation, l'entraînement des différents opérateurs humains ainsi que leur sensibilisation sont essentiels. Cela permet aux employés d'acquérir les connaissances nécessaires pour utiliser correctement les systèmes, se conformer aux politiques de sécurité et manipuler les données.

Soutien du Management La réussite d'une culture de sécurité dans une organisation passe également par un bon management et un bon leadership. En effet lors des gestions de crise, il va falloir appliquer un management cohérent afin de mettre en œuvre les procédures identifiées lors des entraînements. Un bon leadership est aussi nécessaire pour éviter un chaos et permet de prendre des décisions de manière claire dans des situations complexes.

4.1.2 Outils conceptuels existants

Il existe plusieurs méthodes et outils visant à s'intéresser au facteur humain dans le développement sécurisé d'un système. Nous en présentons trois ici.

RIDIM Il s'agit d'une méthode de management de risques pour les organisations [43]. RIDIM prend en compte des concepts venant de l'ingénierie des exigences et y ajoute des politiques de sécurité applicables au milieu de l'entreprise. Cette méthode a été créée en s'appuyant sur un certain nombre d'éléments déjà connus comme SWOT [44]. SWOT est un outil de gestion de stratégie d'entreprise qui

permet d'identifier les forces et les faiblesses d'une entreprise à travers la prise en compte d'éléments internes à l'entreprise ainsi que d'éléments externes. Cet outil permet au final de prendre des décisions dans un domaine stratégique qui est liée à l'entreprise. SWOT a été modifié afin d'y incorporer un modèle de facteur humain. Ce dernier est composé de facteurs directs représentant tous les facteurs représentant directement un opérateur humain et également de facteurs indirects qui sont la représentation de tous les facteurs environnementaux pouvant influencer l'individu. Voici les différents facteurs appartenant aux familles de facteurs directs et indirects :

Facteur direct : l'erreur, la sensibilisation, les compétences, l'expérience, l'apathie, l'ignorance et la négligence, le stress.

Facteur indirect : le budget, la culture, la communication, le soutien de la direction, l'application de la politique de sécurité, la politique d'incitation et de dissuasion.

Cette approche est aussi complétée par des éléments venant directement de l'ingénierie des exigences lui permettant de définir les différents processus et objectifs de l'organisation pour y placer un certain nombre de risques potentiels à travers notamment le facteur humain

Au final, RIDIM permet à l'entreprise de venir calculer, en fonction des différents acteurs et objectifs de l'organisation, le retour sur investissement possible des mesures de sécurité mises en place sur le facteur humain.

STRIDE-HF L'approche proposée [45] est récente. Les auteurs proposent un Framework centré sur la modélisation de la menace à travers le facteur humain. Pour ce faire, elle s'appuie sur le Framework STRIDE déjà existant [46] qui permet de modéliser des menaces cyber sur les systèmes. Les auteurs ont ajouté à STRIDE des notions de facteur humain pour former un nouveau Framework : STRIDE-HF. Ce dernier offre aux analystes en cybersécurité un moyen de prendre en compte le comportement lié au facteur humain tout en évaluant les types d'attaque qui pourraient en résulter. Pour ce faire le framework s'appuie sur la classification des attaques suivante (nous donnons le nom des attaques en français et en anglais car les deux sont communément utilisés) :

Usurpation d'identité (*Spoofing*) Cela consiste à utiliser les informations d'identification d'une autre personne pour accéder à des ressources autrement inaccessibles.

Falsification (*Tampering*) Il s'agit de modification des données afin de créer une attaque.

Répudiation (*Repudiation*) Cela se produit lorsqu'un utilisateur nie avoir effectué une action, mais que la cible de l'action n'a aucun moyen de prouver le contraire.

Divulgation d'informations(*Information disclosure*) Un opérateur humain va divulguer des informations à un tiers qui n'est pas censé y avoir accès.

Déni de service (*Denial of service*) Cette attaque vise la réduction de la capacité des utilisateurs d'un système à accéder aux ressources de celui-ci.

Élévation de privilège (*Elevation of privilege*) Cela se produit lorsqu'un utilisateur non privilégié obtient un statut privilégié.

À cette classification, les auteurs ont ajouté les différents comportements humains pouvant correspondre à ce type d'attaque. Cette approche permet à l'utilisateur du langage d'identifier le type de vulnérabilité classifié dans STRIDE qui correspond au facteur humain. Une fois identifié, l'utilisateur peut mettre en place de nouvelles mesures visant à éviter ce risque. On peut regretter que le facteur humain considéré par l'approche ne se limite pour le moment qu'à une description des comportements que l'on va chercher à faire correspondre à de potentielles attaques. Comme l'annoncent les auteurs, il reste encore un certain nombre d'éléments à formaliser pour permettre d'avoir une évaluation plus fine de la vulnérabilité liée aux facteurs humains.

Le framework de [47] Les auteurs de cet article proposent un Framework permettant de prendre en compte le facteur humain dans la culture de la sécurité informatique d'une organisation. Pour ce faire, les auteurs se sont appuyés sur STOP [48], un Framework existant. Ce dernier permet initialement aux dirigeants d'une organisation de comprendre les différents éléments composant la sécurité informatique dans la dite organisation. Pour ce faire, la description de cinq caractéristiques correspondant à cinq étapes sont nécessaire :

La stratégie Ici la notion de stratégie désigne les différentes stratégies de sécurité informatique mises en place dans une organisation. Cela peut prendre la forme de politique de sécurité, d'objectifs de sécurité à atteindre ou de meilleures pratiques.

La technologie Ici le terme technologie renvoie à tous les éléments techniques dans l'organisation. Cela comprend le matériel, les logiciels, les services Web, les applications qui sont utilisées au sein de l'organisation.

L'organisation Ici la notion d'organisation renvoie la structure de l'organisation. Celle-ci peut exercer une forte influence sur la culture de sécurité informatique dans l'organisation.

Les personnes ici le terme de personne renvoie à tout individu prenant part à l'organisation.

L'environnement Ici la notion d'environnement renvoie à tout ce qui est extérieur à l'organisation. Cela peut avoir un fort impact sur elle. Des notions telles que la culture d'un état par exemple ou alors l'existence de différentes normes, qui peuvent s'appliquer à l'organisation et influencer sa sécurité.

Les auteurs ont ajouté à ces cinq étapes une étape supplémentaire. Elle concerne le facteur humain. Cette description du facteur humain est composée de quatre éléments qui vont, à travers les étapes du Framework décrit ci-dessus, permettre de prendre des décisions en fonction des éléments concernant le facteur humain qui peuvent poser des problèmes de sécurité. Ces quatre éléments reliés à la notion de facteur humain sont :

La préparation aux catastrophes Cet élément décrit tout ce qui concerne la formation, la sensibilisation et l'acquisition de connaissances.

La responsabilité Cet élément renvoie à tout ce qui concerne la prise de responsabilité des individus. Cela comprend la surveillance et le contrôle ainsi que la récompense et la dissuasion qui vont mener à l'acceptation de la responsabilité.

Le management Cet élément renvoie toutes les pratiques de management qui sont utilisées sur les individus pour permettre une meilleure politique de sécurité. Cela comprend les pratiques ainsi que les notions de leadership et aussi l'interaction entre les individus.

La société et le réglementation Cet élément s'intéresse à ce qui est principalement lié aux aspects socioculturels et aux questions de réglementations qui peuvent influer sur les individus.

Ce framework permet donc de mettre en place une bonne culture d'entreprise autour de la sécurité avec une prise en compte du facteur humain. Le facteur

humain est ici un élément assez simplifié, mais qui permet malgré tout de faire ressortir certaines actions pouvant influer sur les différents acteurs présents dans une organisation pour limiter leur vulnérabilité.

4.1.3 Discussion

Les langages que nous avons cités ci-dessus représentent chacun une approche différente de la prise en compte du facteur humain dans le but de définir une meilleure sécurité dans des organisations ou des systèmes. La limitation que nous avons pu tirer de ces différents exemples, et plus généralement de la modélisation du facteur humain dans la littérature, est que la manière dont ce dernier est considéré par les différentes approches conceptuelles est souvent trop simplifiée. Cela implique de ne pas permettre la représentation et l'identification de toutes les vulnérabilités qui lui sont liées. Dans certains cas, le facteur humain peut être détaillé, mais alors il n'est pas, à notre connaissance, utilisé pour permettre de détecter ou de simuler la vulnérabilité qu'il peut représenter. De plus, il n'est que rarement utilisé dans des cas complexes qui peuvent générer de l'émergence telles que dans les SoSTS. Il apparaît donc nécessaire de développer une approche permettant une prise en compte significative du facteur humain dans les SoSTS avec pour objectif d'identifier les vulnérabilités possibles pouvant être engendrée dans le système. Il va falloir pour cela caractériser ce qu'est effectivement un modèle humain, définir la vulnérabilité humaine et l'impact de ce facteur humain sur le système.

4.2 Estimation de la vulnérabilité humaine

L'action qui consiste à estimer la (ou les) vulnérabilité(s) d'un système peut s'envisager à différents niveaux. On peut par exemple estimer la vulnérabilité d'un logiciel, d'un système ou d'une organisation. L'objectif derrière cette action est de mettre en évidence les vulnérabilités ou les défauts liés à ce qu'on étudie (logiciel, système ou organisation) pour permettre son amélioration. L'estimation de la vulnérabilité peut se mener sur un élément existant ou sur un élément que l'on cherche à concevoir. Elle se différencie des recherches de vulnérabilité par tests, en ce sens qu'elle s'attache à estimer les vulnérabilité à partir de modèles conceptuels, y compris sur des systèmes ou organisation existants.

Nous nous concentrerons sur les méthodes d'analyse permettant d'estimer des vulnérabilités de sécurité informatique dans les systèmes ou organisations. Nous les regroupons

selon deux types d'approche, en fonction de l'usage ou non d'un outil de simulation, ou si seule l'expertise d'un expert est demandée. La première approche est dite "l'approche par expertise", dans laquelle un expert va, grâce à son expérience et son expertise, via l'usage ou non d'outils, chercher à mettre en évidence des vulnérabilités dans les systèmes et/ou les organisations. La deuxième approche est une approche outillée qui va utiliser la simulation et des modèles de vulnérabilité pour essayer de déduire des vulnérabilités potentielles en fonction d'une architecture donnée.

4.2.1 Approche par expertise

L'approche par l'expertise est la plus employée même de manière indirecte. En effet, lorsque l'architecte définit l'architecture d'un système ou d'une organisation, il utilise son expertise pour permettre au mieux à cette architecture d'être à la fois efficiente et correspondante aux besoins exprimés. Dans le cas d'un architecte ayant une connaissance en termes de sécurité informatique, il va nativement rendre plus résistante cette architecture en palliant un certain nombre de vulnérabilités dès cette phase de conception. Dans ce contexte, l'expertise de l'architecte peut venir de sa propre expérience ou de formations qu'il a reçues.

Cependant, ce type de sécurisation n'est pas toujours suffisant. En effet, les architectes possédant une expérience significative en termes de sécurité sont souvent rares d'une part, et d'autre part, il est parfois complexe de conjuguer la double expertise. Enfin, la rapidité avec laquelle la cybersécurité évolue nécessite une mise à jour des compétences régulière. Tout cela concourt à rendre insuffisante cette approche de sécurisation. Aussi, des outils sont venus soutenir les architectes afin de mieux isoler les risques de vulnérabilité de manière à, soit les corriger, soit les diminuer.

Par exemple, la méthode EBIOS [49], a été développé à partir de 1995 par la direction centrale de la sécurité des systèmes d'information et est actuellement maintenu par ANSSI. Dans sa dernière version, le but est d'aider les experts à prendre en compte les risques liés à la cybersécurité à travers cinq étapes appelées ici ateliers.

Au final, l'approche que propose EBIOS permet la prise en compte des vulnérabilités dans le système ou dans l'organisation. Cependant, la réalisation de cette méthode requiert une expertise pour être mise en place efficacement. De plus, malgré l'expertise, il peut rester des angles morts. En effet, si l'on oublie des éléments, que ce soit dans les risques ou les personnes qui peuvent prendre part aux différents scénarios, cela peut biaiser les résultats.

4.2.2 Approches outillées

Dans ce deuxième type d'approche, plusieurs sous-catégories peuvent être identifiées. Nous pouvons énoncer l'extension de l'approche par expertise qui va, à travers un outillage, utiliser des modèles définis par des experts afin d'évaluer des risques ou des vulnérabilités. Ce type d'approche va appliquer une vision de vulnérabilité très arithmétique dans son estimation et s'adaptera peu aux architectures qui lui seront présentées.

Parmi les sous-catégories de cette approche entrant dans notre champ d'étude, il y a les approches par apprentissage. Elles s'appuient sur des données existantes pour amener des améliorations au modèle de vulnérabilité défini, ou pour déduire un modèle de vulnérabilité correspondant aux données disponibles et collant au plus près du système.

Toutes ces approches n'ont été appliquées que récemment au monde de la cybersécurité [50]. En effet, la communauté n'a commencé à s'intéresser à l'usage de ces approches qu'à compter des années 2010. Leur utilisation va crescendo depuis. Ces approches par apprentissage sont très dépendantes des données qu'elles manipulent. Ce sont en effet les données d'entrées qui vont leur permettre de développer des conjectures permettant l'estimation de la vulnérabilité. On peut classer les approches en deux familles en fonction de la nature des données, en fonction de la capacité à avoir des données correspondant à des résultats complets, ou alors des données indirectes permettant une déduction des résultats.

4.2.2.1 Approche avec données existantes

Dans la famille des approches utilisant des données complètes, il existe plusieurs algorithmes d'apprentissage souvent qualifiés d'« intelligence artificielle. » Nous pouvons par exemple citer [51] où les auteurs ont utilisé un réseau neuronal profond (DNN - DNN deep neural network). Les auteurs ont alimenté le DNN à partir d'un jeu de données nommé *KDD Cup 99*. À partir de ces données qui sont labelisées, le réseau va apprendre à reconnaître des schémas d'attaque dans les réseaux pour pouvoir détecter ces attaques sur de nouveaux réseaux. Les auteurs estiment qu'avec un apprentissage sur ce jeu donné, le DNN arrive à de bons résultats de détection d'attaques dans des réseaux informatiques. Ce travail est illustratif des limites de ce type d'approche. En effet, les auteurs se basent sur un jeu de données qui date de 1999. Cela signifie que l'approche ne saura reconnaître que des attaques dont le schéma a été construit sur un mode opératoire datant d'avant 1999, ce qui représente la préhistoire en terme de cybersécurité. Des schémas d'attaque

plus récents ne seront pas détectés et cela va générer un certain nombre de biais lors de l'application de cette approche sur des réseaux actuels [52].

De plus, l'évolution des jeux de données montre qu'ils sont de plus en plus spécifiques suivant ainsi la montée en puissance de la numérisation des systèmes. On trouve en effet maintenant des jeux de données spécifiques par exemple à des systèmes industriels [53] ou encore à l'Internet des objets connectés [54].

Cela nous amène à conclure qu'une approche par apprentissage profond peut être très performante dans certains cas, mais elle a besoin d'être construite sur des jeux de données adaptés et récents, plus généralement sur des jeux de données correspondant au besoin de détection ou d'estimation. À notre connaissance il n'y a pas à ce jour de tels jeux de données existants pour identifier la vulnérabilité humaine dans des systèmes de systèmes socio-techniques.

4.2.2.2 Approche sans données existantes

Le fait de ne pas avoir de données préétablies pour permettre un apprentissage vis-à-vis d'un modèle de vulnérabilité est un élément qui peut poser des difficultés. Aussi, afin de pallier à ce problème, la plupart des approches vont subdiviser le problème qu'elles cherchent à modéliser. Le modèle final sera composé de toutes ces subdivisions pour permettre de faire une conjecture vers la solution finale. Afin d'illustrer cette approche, détaillons celle proposée par [55]. Les auteurs mettent en place un modèle d'analyse des risques et de sécurité pour permettre d'évaluer le risque propre au système d'information complet d'une organisation. Comme il n'existe pas de telles données, les auteurs proposent une approche avec un réseau bayésien permettant, à partir de données de sous-composants du système d'information, d'estimer la vulnérabilité de ce système d'information.

Cette approche utilisant des modèles probabilistes, et plus spécifiquement des réseaux bayésiens, est utilisée dans d'autres domaines que celui de la cybersécurité. Dans [56] par exemple, les auteurs utilisent les réseaux bayésiens pour estimer la vulnérabilité d'individus humains face à l'environnement et plus spécifiquement aux tremblements de terre.

4.2.3 Discussions

Les différentes approches que nous avons pu présenter, pour l'estimation de la vulnérabilité présentent cependant des limites vis-à-vis de l'estimation de la vulnérabilité du facteur humain dans un SoSTS. Il n'existe pas en effet de modèle préétabli d'estima-

tion de la vulnérabilité humaine dans le monde de la cybersécurité. Cependant il existe des éléments proches, notamment autour des réseaux bayésiens qui permettent, en plus d'aborder le problème en le décomposant sous différents angles et en les liant entre eux, de ne pas avoir à gérer la problématique de l'actualisation des données. Aussi pour répondre à ce manque, il paraît nécessaire de développer une approche de simulation permettant d'estimer la vulnérabilité humaine.

Conclusion de partie

Concevoir un SoSTS sans la prise en compte de la vulnérabilité humaine pose un problème en termes de cybersécurité. Comme nous l'avons montré dans cette partie, la conception d'un SoSTS est déjà à l'origine d'une complexité accrue puisqu'elle doit composer avec des éléments issus du monde SoS et les éléments venant du monde des ST. Tous ces éléments forment ensemble les SoSTS et permettre la modélisation de systèmes intégrant à la fois les systèmes techniques et humains, le tout en intégrant l'émergence de comportements issus de l'interaction de ces différents éléments.

Nous avons ensuite présenté le domaine de la sécurité par conception. Pour permettre une conception sécurisée dans le cadre des SoSTS qui est un domaine récent, nous nous sommes penchés sur ce qui existe du côté des SoS et des ST. Nous avons pu, à travers cela, faire émerger notre première question de recherche **QR1**.

Ensuite, nous avons abordé la façon dont la modélisation de l'humain pouvait être prise en compte de manière à permettre l'estimation de vulnérabilités. Nous avons montré que les éléments de littérature existants ne permettent pas de modéliser la vulnérabilité humaine dans un SoSTS. Nous avons cependant pu présenter un certain nombre d'éléments existants utiles. Cela nous a permis de faire émerger notre deuxième question de recherche **QR2**.

Pour ce qui est de l'estimation de la vulnérabilité humaine, plusieurs pistes existent. L'élément déterminant, comme souvent, se limite à l'expertise d'un individu d'une part, ou d'autre part à l'utilisation de données permettant de faire des conjectures sur un modèle représentant l'humain, ce qui amène à une estimation de la vulnérabilité humaine. Cette problématique d'estimation est directement reliée à notre question de recherche **QR3**.

Prendre en compte la vulnérabilité humaine implique d'avoir un moyen d'intégrer dans la conception du système des éléments nécessaires à son estimation. Il faut pour cela définir les éléments caractérisant l'humain de manière à permettre l'estimation de sa vulnérabilité. Différentes approches existent, mais ne permettent pas toutes de prendre en compte la vulnérabilité associée au facteur humain et son intégration dans les SoSTS.

Troisième partie

Contributions

Chapitre 5

Modéliser l'humain dans un SoSTS

Sommaire

Le chapitre précédent a montré le manque de supports adaptés à la prise en compte de la vulnérabilité humaine dans les modèles d'architectures socio-techniques. Dans ce chapitre, nous proposons notre première contribution visant à pallier ce manque. Elle consiste à fournir un moyen de décrire une architecture SoSTS en prenant en compte l'humain et surtout ses caractéristiques. Pour ce faire, il faut dans premier temps caractériser la manière dont un humain doit être représenté dans une telle approche.

Nous proposons pour cela un modèle du facteur humain permettant d'estimer la vulnérabilité humaine à la section 5.1. Ce modèle est bâti autour d'un modèle conceptuel décrivant des propriétés qui caractérisent à la fois l'humain et son environnement, que l'on appelle facteurs. Sont également prise en compte les relations entre humains. Les facteurs considérés sont ceux nécessaires à l'évaluation de la vulnérabilité humaine vis-à-vis d'un SoSTS. Ils sont décrits par la suite. Enfin, le modèle spécifie la manière de quantifier ces différents facteurs. Dans un deuxième temps, notre démarche cherchant à se situer

dans un contexte d'ingénierie, nous définissons dans la section 5.2 un langage de modélisation permettant de manipuler le modèle humain proposé. Enfin, de manière à rendre fonctionnel le langage défini, et de manière à en valider les concepts, nous proposons une implémentation de ce langage à la section 5.3. Le travail de recherche qui est présenté ici fait l'objet de plusieurs publications [57], [58].

5.1 Un modèle de facteur humain

L'approche que nous défendons dans cette thèse est de modéliser, dans les phases amonts de la conception d'un SoSTS, les opérateurs humains afin d'évaluer leur vulnérabilité. Ainsi, dans cette section, nous nous intéressons à définir un moyen permettant de représenter un humain évoluant dans un SoSTS. De nombreux travaux dans le domaine des sciences sociales et politiques ont conduit à la conclusion que, telle la plupart des systèmes naturels complexes, il est illusoire d'espérer représenter de manière complète (au sens mathématique du terme) l'humain. Partant de ce constat, il convient donc de réduire la complexité du système humain de manière à n'en représenter qu'un sous-ensemble représentatif et interprétable. Cela est rendu possible grâce au mécanisme de l'abstraction.

Dans ce contexte, il convient de s'interroger sur la nature du sous-système à représenter, et plus précisément sur la (ou les) propriété(s) de ce sous-système à modéliser. En effet, ici nous nous intéressons à la vulnérabilité humaine. Ainsi, il convient d'identifier quelles caractéristiques du système humain sont impactantes sur sa vulnérabilité, et, le cas échéant, d'étudier si les relations entre les opérateurs peuvent avoir un impact sur cette même vulnérabilité.

Pour ce faire, nous présentons en premier lieu le métamodèle de l'humain que nous proposons et qui permet de caractériser un opérateur humain dans un SoSTS. Cette caractérisation repose notamment sur l'identification de propriétés propres à l'opérateur mais aussi les influences extérieures qui peuvent le concerner. Nous donnerons donc dans deuxième temps, la liste des propriétés humaines que nous avons identifiées et que nous jugeons impactantes pour évaluer la vulnérabilité humaine d'un opérateur, ainsi que la liste des propriétés extérieures pouvant influencer sa vulnérabilité. Enfin, une fois ces propriétés identifiées, il convient de déterminer un moyen de les évaluer. Pour cela, nous proposerons une manière de les quantifier.

5.1.1 Métamodèle de l'humain à travers ses facteurs

Comme mentionné précédemment, caractériser un opérateur humain de manière à en évaluer la vulnérabilité au sein d'un SoSTS consiste à définir un certain nombre de propriétés le concernant puis à raisonner par la suite dessus. En effet, modéliser l'humain dans toute sa complexité est à ce jour impossible. C'est également non pertinent. Autant certaines propriétés, telles que par exemple le niveau de formation, ont un intérêt certain pour évaluer cette vulnérabilité, autant d'autres, telles que la couleur des yeux, n'ont pas d'intérêt objectif. Ainsi, modéliser l'opérateur humain avec pour objectif d'évaluer sa vulnérabilité implique d'identifier les propriétés humaines pouvant être sources de vulnérabilité. Nous parlerons par la suite de "*facteurs humains*".

Cette idée de caractériser un opérateur humain en utilisant un certain nombre de facteurs dédiés afin d'évaluer la capacité du-dit opérateur à répondre à des situations spécifiques est inspirée des travaux de [43]. Les auteurs ont proposé de créer des modèles dans lesquels la vulnérabilité humaine dépendait de facteurs directs et indirects. Les facteurs directs permettaient de caractériser un individu humain et cela indépendamment de toute influence extérieure. Les facteurs indirects correspondaient aux éléments extérieurs pouvant avoir une influence sur l'individu. Ces éléments extérieurs sont par exemple la tâche qui est assignée à l'opérateur ou le type de management qui va s'appliquer à l'opérateur lors de la réalisation de cette tâche.

Comme nous l'avons détaillé dans le précédent chapitre, ce travail entre dans le contexte de la description d'une méthode de management de risques pour les organisations. Il ne s'agit donc pas du même cadre que nos travaux. Cependant nous jugeons cette approche particulièrement pertinente. Nous nous en inspirerons dans la suite de ce travail.

Les travaux de [43] permettent une première identification des différents facteurs directs et indirects à prendre en compte. Une rapide analyse des facteurs directs et indirects pris en compte par [43] montre qu'un certain nombre n'ont pas forcément d'intérêt dans le cadre de notre problématique. Ce constat est naturel. On peut comprendre que lorsqu'un individu est fatigué il va commettre plus d'erreurs. Cela peut aussi augmenter son temps de réaction, ce qui pourrait être dangereux dans une situation particulière. On peut aussi supposer que lorsqu'une situation stressante s'exerce sur des individus, certains peuvent perdre leur capacité à évaluer correctement celle-ci et donc potentiellement prendre une mauvaise décision. Les facteurs a prendre en compte pour évaluer la capacité d'un opérateur humain à évoluer dans un contexte spécifique sont donc dépendant à la fois de ce

même contexte, mais aussi des caractéristiques que l'on cherche à évaluer. Ainsi, nous proposons de définir dans un premier temps un modèle générique de l'opérateur humain, de ses facteurs directs et des facteurs indirects qui le concernent dans un contexte particulier.

On peut également noter le manque de caractérisation claire des influences existant entre les facteurs directs et indirects. Nous pouvons illustrer ces relations d'influence par l'exemple suivant : imaginons un opérateur humain, qui peut être stressé rapidement au quotidien et faire des erreurs liées à ce stress. Il peut être canalisé par un encadrement adapté venant de son manager. Cela peut permettre de diminuer les erreurs imputables à cet opérateur. Cet exemple illustre un lien qui est documenté dans la littérature et peu pris en compte par [43]. Il s'agit de l'influence du management sur la stabilité émotionnelle d'un individu comme nous le verrons à la section suivante. Il y a donc un besoin à caractériser les relations entre facteurs directs et indirects lorsqu'elles existent. Permettre cette formalisation doit donc être possible dans notre approche.

Une autre limitation que nous avons identifié est que les modèles proposés dans ces travaux sont principalement utilisés pour représenter une situation, mais pas pour effectuer des traitements dessus. Or le traitement nécessite d'être capable de quantifier les propriétés sur lesquelles l'analyse est menée, ce qui n'est pas le cas. Dès lors, notre approche qui cherche à évaluer automatiquement une vulnérabilité, aura nécessairement ce besoin. Il faut donc définir un moyen de quantifier les facteurs humains. Or, si certains facteurs peuvent être facilement quantifiés par des échelles de valeurs (le niveau de formation par exemple), d'autres nécessitent une représentation plus aisément interprétable. Par exemple, c'est le cas de la stabilité émotionnelle qui est plus aisément compréhensible à l'aide d'adjectifs : stable, anxieux, ... Nous reviendrons plus en détails sur ces deux familles de quantifieur dans la Section 5.1.3. Mais nous retenons pour le moment le fait, qu'un opérateur humain peut être considéré à travers l'ensemble de ses facteurs directs et indirects. Ces facteurs peuvent être quantifiés par des valeurs pouvant être de type numérique ou adjective.

Dans l'évaluation de la vulnérabilité humaine d'un opérateur humain, il convient également de s'intéresser aux relations entre opérateurs. En effet, les opérateurs interagissant au sein d'un SoSTS ne sont pas seulement entourés de systèmes. Ils ont besoin de collaborer avec d'autres individus pour la réalisation de leurs tâches, en plus de la collaboration avec les systèmes techniques nécessaires à la réalisation de ces même tâches. Les relations entre individus sont essentielles dans une modélisation humaine, puisque l'humain est un être social, il va être influencé par des individus voire par des mécaniques de groupe [59].

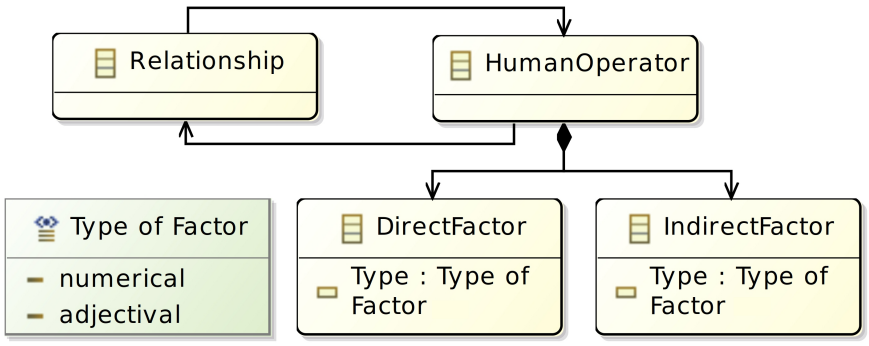


FIGURE 5.1 – Métamodèle de l'humain à travers ses facteurs

Prendre en compte les relations entre opérateurs est donc une nécessité.

Ainsi, de manière à spécifier le cœur de notre approche, nous proposons de définir un métamodèle à partir des remarques précédentes. L'intérêt est de spécifier un certains nombre de points importants dont nous venons de discuter afin de pouvoir décrire dans un modèle un SoSTS et d'appliquer des analyses à ce modèle. La figure 5.1 représente ce métamodèle. L'opérateur humain est caractérisé par une métaclasse `HumanOperator`. L'opérateur doit être caractérisé en termes de facteurs humains directs et indirects. Afin d'offrir de la généricité, et ainsi de pouvoir utiliser ce métamodèle dans différents contextes, nous ne donnons pas de liste de facteurs directs (`DirectFactor`) et indirects (`Indirect Factor`). Ils pourront ainsi être spécifiés en fonction d'un contexte donné. Ces facteurs doivent pouvoir être quantifiés. Nous proposons pour le moment de restreindre les types de valeurs à envisager à des valeurs numériques (`numerical`) et adjectives (`adjectival`). L'utilisation d'une énumération (`Type of Factor`) permet d'étendre au besoin ces types. Enfin, nous prenons en compte les relations que peut avoir l'opérateur humain avec la metaclasse `Relationship`. Cette metaclasse est un élément important puisque la vulnérabilité humaine peut se propager à travers des relations entre individus [60]. Cette représentation permet la spécification de toutes les relations qui ne seraient pas représentées dans l'architecture d'un SoSTS. Cela comprend donc les relations qui sont liées à un individu ou les relations qui lui seront données par son rôle dans le SoSTS.

Dans la suite de ce chapitre, nous restreignons notre champ d'étude à la détection de la vulnérabilité humaine. Ainsi, dans la section suivante, nous donnerons le détail des facteurs humains que nous avons identifiés comme pouvant avoir un impact sur celle-ci.

5.1.2 Facteurs humains impactant la vulnérabilité humaine

Les différents facteurs que nous avons identifiés comme pouvant impacter la vulnérabilité humaine sont détaillés ici dans les deux grandes catégories qui sont des facteurs directs et les facteurs indirects.

5.1.2.1 Facteurs directs

Les facteurs directs sont les facteurs qui sont inhérents à la personne elle-même. Cela signifie qu'un facteur direct n'a besoin d'aucun autre élément pour se définir que la personne elle-même. La plupart des facteurs ci-dessous sont des facteurs que nous avons extraits de la littérature, notamment de littérature des sciences humaines. À cela nous avons ajouté des facteurs qui sont issus de la collaboration et de l'expertise de notre partenaire industriel dans le domaine naval et militaire. La plupart de ces facteurs ont été sélectionnés en fonction de l'impact qu'ils pouvaient avoir sur la vulnérabilité humaine d'une part et d'autre part sur leur lien avec le domaine de la cybersécurité.

Compétence : La compétence se définit comme étant un facteur qui représente aussi bien la formation d'un individu que son éducation. La compétence est reconnue comme un élément majeur dans la gestion de crises [61]. En effet, lorsqu'un individu est correctement formé il peut avoir les bons réflexes et éviter un incident. De plus, la compétence est essentielle pour certains postes. En effet, si le niveau de compétence minimale n'est pas atteint, cela peut constituer *de facto* une vulnérabilité. On peut prendre pour exemple un opérateur de chaudière nucléaire dans un sous-marin nucléaire. Il est évident que sans formation adéquate, le risque que l'opérateur, dans la réalisation de sa fonction, conduise à une situation dangereuse est très élevé.

La compétence d'un individu peut évoluer au fur et à mesure du temps. Elle peut par exemple décroître suite à l'évolution de matériels et/ou de procédures. Il est donc important de maintenir la compétence individuelle a travers des formations par exemple pour éviter que celle-ci ne génère une vulnérabilité dans le système.

Expérience : L'expérience caractérise la connaissance d'un poste donné (ou d'une fonction donnée) en fonction du temps que celui-ci y a passé [62]. Ce facteur est complémentaire à la compétence. Il ne se substitue pas à la connaissance théorique d'un métier, mais vient renforcer cette connaissance avec de la connaissance opérationnelle. Cela indique une meilleure capacité d'un individu à remplir pleinement

sa fonction.

Ce facteur peut être une source de vulnérabilité si un individu n'est pas suffisamment expérimenté pour occuper un poste dans lequel un niveau d'expérience donné est demandé. En effet, certains postes demandent plus que la simple formation nécessaire pour y avoir accès : pour les occuper efficacement, il peuvent nécessiter des années d'expérience pour permettre une efficacité de l'opérateur sans que cela ne génère de vulnérabilités potentielles. Par exemple, une personne qui serait responsable de toute une infrastructure de systèmes critiques ne doit pas simplement connaître la théorie sur chacun de ces systèmes. Pour être efficace, elle doit avoir expérimenté l'usage de chacun de ces systèmes pour en connaître leurs faiblesses. Ainsi, lors d'un incident sur l'un des systèmes sous sa responsabilité, l'intervenant pourra réagir plus efficacement. Il réduit ainsi les vulnérabilités potentielles.

Fiabilité : La fiabilité se définit comme la propension d'un sujet à ne pas commettre d'erreur dans le poste qu'il occupe [63]. Ce facteur permet de mettre en évidence les comportements qui peuvent faire naturellement naître de la vulnérabilité. Une fiabilité extrêmement forte peut être en effet demandée à un poste critique. De plus, même sur certains postes où la fiabilité n'est pas une nécessité absolue, elle peut en cas de crise être un facteur aggravant, voir s'additionner avec d'autres facteurs, pour augmenter la vulnérabilité d'un individu. La fiabilité d'un individu peut-être également éprouvée par le système lui-même. En effet, plus le système demande des interactions complexes avec un individu, plus il y a de chances que cet individu commette des erreurs. La fiabilité d'un individu peut donc aussi évoluer au fur et à mesure du temps et en fonction des tâches qu'il va réaliser.

Conscience : Ce facteur permet de prendre en compte la capacité d'un opérateur humain à s'investir dans la tâche qui lui est confiée. En effet, un individu peut avoir plus ou moins envie de s'investir dans son travail : cet investissement peut avoir un réel impact sur la qualité des tâches effectuées [64]. Ce facteur peut être ainsi source de vulnérabilité. Si une personne est trop peu consciencieuse envers son travail, elle pourra, par exemple, ne pas terminer ou ne pas faire correctement une tâche (pour, imaginons, gagner du temps). Cela pourrait introduire potentiellement une vulnérabilité. Prenons par exemple le fait qu'un opérateur doit analyser les logs d'un ordinateur afin d'en vérifier la sécurité informatique. Cette tâche peut-être répétitive et surtout cet opérateur a l'habitude de l'effectuer. Il peut être tenté de ne regarder que rapidement les logs sans faire trop attention. Si cela se produit,

alors il pourrait passer à côté d'une vraie alerte de sécurité et ainsi aurait introduit une vulnérabilité lors de procédures qui auraient pourtant pu identifier une menace.

Confiance : La confiance est l'un des facteurs que nous avons identifié en s'appuyant sur notre partenaire industriel. il représente ici le niveau de confiance que l'organisation accorde à un opérateur qui occupe un poste au sein du SoSTS. C'est un facteur qui est important dans la définition de la sécurité. En effet, la confiance que l'on donne à un individu qui va occuper un poste va refléter les différentes informations et objectifs sensibles que va manipuler cet individu. Prenons par exemple un individu qui détient des informations sensibles pour le SoSTS dans lequel il évolue. Si ces informations venaient à être compromises, cela pourrait avoir un impact important sur le système lui-même.

Robustesse : La robustesse est un facteur également identifié en lien avec l'expérience de notre partenaire industriel. Il sert à représenter la capacité physique d'un individu occupant un poste au sein d'un SoSTS. Ce facteur représente notamment la capacité des individus à encaisser de forts moments de stress et de crise dans un SoSTS. En effet, certains postes peuvent exiger une plus grande force physique ou mentale. Dans le cas où l'opérateur qui occupe ce poste n'a pas cette capacité de robustesse, cela pourrait constituer une vulnérabilité sur l'individu. Considérons par exemple le cas des pilotes de chasse. Dans leur cas, la fatigue mentale qui est générée par la charge cognitive, ainsi que la fatigue physique liée à l'encaissement des mouvements de l'avion, sont des éléments qui sont particulièrement importants lors du pilotage. Un pilote de chasse qui serait alors en condition physique et mentale déjà dégradée avant de commencer son vol pourrait créer une vulnérabilité importante.

Niveau informationnel : Ce facteur permet de prendre en compte la capacité d'un opérateur humain à ne pas ignorer les procédures, que ce soient les procédures liées aux tâches qu'il doit réaliser, ou bien des procédures liées à la politique de sécurité de l'organisation [65]. Ce facteur est en effet source de vulnérabilité si la personne est trop peu informée et donc va naturellement ignorer certaines politiques de sécurité . Prenons par exemple la politique de sécurité sur les mots de passe que la personne va ignorer, notamment due à un manque de formation. Ce comportement va de fait induire une vulnérabilité dans le système. Ce facteur illustre un comportement qui est souvent responsable d'attaques réussies dans les SoSTS. Par

ailleurs, il faut bien différencier ce facteur du facteur Fiabilité. En effet, ce facteur-ci cherche surtout à décrire l'attitude d'une personne qui aurait tendance à vouloir contourner les procédures par manque de connaissance.

Coopération organisationnelle : Ce facteur représente la capacité d'un individu à adhérer aux valeurs d'une organisation [66] : plus un individu est en disposition d'adhérer à l'organisation ainsi qu'aux valeurs portées par celle-ci, moins il sera réticent à appliquer les procédures ainsi que les directives de cette organisation. Ce facteur permet de prendre en compte également la capacité d'un individu à coopérer plus localement avec les autres opérateurs et systèmes techniques qui l'entourent. Par exemple, un militaire représenterait une vulnérabilité si celui-ci était en constante opposition avec les positions portées par l'armée à laquelle il appartient. Il en serait de même s'il contestait systématiquement l'application des ordres reçus

Stabilité émotionnelle : Il s'agit de la capacité d'un individu à contrôler ses émotions dans une situation pouvant les perturber. Il s'agit d'un élément important dans la modélisation de l'humain puisque dans certaines situation, une stabilité émotionnelle ne correspondant pas au poste occupé peut générer une vulnérabilité dans le système [67]. Les cas de gestion de crises face à une menace illustrent bien la nécessité, pour les opérateurs impliqués dans une situation donnée, de rester sur un niveau d'émotion stable. De plus, certains rôles nécessitent une stabilité émotionnelle particulièrement importante ; par exemple, on peut noter le cas des pilotes d'avion ou des contrôleurs aériens qui doivent garder leur sang-froid malgré de situations pouvant engendrer un stress très important.

La majeure partie des facteurs identifiés ci-dessus sont issus de la littérature des sciences sociales. Les facteurs confiance et robustesse sont des facteurs que nous avons fait émerger en lien avec notre partenaire industriel suite à la consultation de retours d'expérience réels. Ils ne bénéficient pas à ce jour d'une assise scientifique comme les autres, mais leur prise en compte a été jugée nécessaire par les experts industriels. En cela, ils montrent bien l'intérêt d'avoir une approche adaptable au contexte, permise par le métamodèle proposé. Ainsi, en fonction du contexte industriel, ou d'un domaine applicatif particulier, il est possible d'adapter le modèle de facteurs humains.

5.1.2.2 Facteurs indirects

Après avoir donné la liste des facteurs directs, c'est à dire des facteurs caractérisant l'opérateur humain en tant que tel, nous dressons ici la liste des facteurs indirects. Ces derniers représentent les éléments environnementaux interagissant avec l'opérateur. Issus de la littérature, ces facteurs sont comme pour les facteurs directs orientés autour de la cybersécurité.

Management : Ce facteur représente assez bien ce que peut être l'influence d'un facteur indirect sur l'opérateur humain. En effet, il permet de caractériser le type de management auquel est soumis l'opérateur humain considéré. En effet, le type de management pour avoir une grande influence sur la capacité d'un individu à être affecté par le stress inhérent à son travail [68]. En cybersécurité, le stress est un élément important pouvant jouer sur la vulnérabilité d'un individu. Nous avons déjà abordé l'importance de cet élément, notamment à travers le facteur direct de la stabilité émotionnelle. Intégrer ce facteur dans notre modèle permet de représenter l'influence que peut avoir l'environnement professionnel sur l'opérateur, car ce facteur permet non seulement de diminuer la vulnérabilité humaine lorsque le management correspond à la personnalité d'un opérateur, mais il peut aussi dans le cas contraire l'aggraver.

Politique de sécurité : Ce facteur représente le niveau des bonnes pratiques de sécurité requise par les tâches ou un poste que va occuper l'opérateur [69]. Le niveau de sécurité est un élément mécaniquement important lorsqu'il s'agit de représenter de potentielles vulnérabilités dans un SoSTS. Par exemple un niveau de politique de sécurité faible sur un poste signifie d'une part qu'il n'y a pas d'éléments sensibles dans ce poste et d'autre part que les procédures liées à la cybersécurité dans ce poste ne sont donc pas primordiales. À l'inverse un rôle présentant un niveau de sensibilité élevée demandera naturellement un niveau de politique de sécurité beaucoup plus élevée afin d'éviter l'introduction de vulnérabilité technique par l'opérateur.

Culture : Ce facteur désigne le type de culture d'entreprise dans lequel va évoluer l'opérateur. La culture d'entreprise peut favoriser l'émergence de vulnérabilités lorsque l'organisation porte des valeurs qui ne correspondent pas à celles pour lesquelles l'opérateur pourrait adhérer [70]. Nous pouvons illustrer ce facteur en prenant l'exemple de la culture d'entreprise présente dans une organisation militaire

qui est une culture d'entreprise très particulière. Elle est profondément pyramidale. Cela peut donc avoir des effets positifs notamment sur la gestion hiérarchique, mais peut avoir des effets négatifs notamment dans l'apparition de vulnérabilité individuelle lorsque l'opérateur est réticent à ce type de culture d'entreprise. Dans ce type d'organisation, c'est souvent lors des moments de crise que la vulnérabilité de l'individu se révélera lorsque par exemple, il ne sera pas capable d'adhérer au groupe dans les instants les plus critiques.

Communication : Ce facteur illustre le type de communication requise sur un poste. La communication peut en effet favoriser l'émergence de vulnérabilité si l'opérateur ne gère pas correctement le type de communication qui lui est demandé [71]. Par exemple, un opérateur ayant une communication active dans certains contextes peut conduire à une compromission d'informations sensibles qui pourraient mettre en danger toute l'organisation. À l'inverse, l'absence de communication dans un environnement où cela est nécessaire peut compliquer les situations et créer de la vulnérabilité supplémentaire ; on peut donner comme exemple les postes de travail où il est nécessaire de gérer des situations de crise ou encore ceux dans lesquels il faut gérer du facteur humain. Dans ce cadre, l'absence de communication a souvent un aspect aggravant.

Exigence de la tâche : Ce facteur représente certains éléments dont une tâche a besoin pour être réalisée dans de bonnes conditions, comme par exemple la contrainte de temps ou la complexité de la tâche [72]. Ce facteur peut favoriser la vulnérabilité lorsque l'opérateur ne peut remplir les différentes conditions permettant de mener à bien la tâche. Par exemple si un opérateur met trop de temps pour réagir à une situation dans laquelle une réactivité importante est requise, cela peut avoir des conséquences tragiques sur le poste qu'il occupe et/ou le SoSTS. Ce facteur peut aussi favoriser la vulnérabilité lorsque la tâche exerce une grande pression sur le physique de l'opérateur ou bien sur son mental. En effet, ces contraintes peuvent alors à terme fatiguer l'opérateur réduisant sa capacité à réaliser correctement son travail et potentiellement permettre ainsi à une menace d'exploiter cette vulnérabilité pour atteindre le système.

Ressource : Ce facteur représente les ressources mises à disposition de l'opérateur occupant un poste. Ce facteur peut en effet aider à la réalisation de la tâche et son absence peut être source de vulnérabilité. En effet, lorsqu'une situation nécessite de plus grandes ressources humaines ou techniques voire monétaires, il faut que l'opé-

rateur réalisant cette tâche ait la capacité de pouvoir les mobiliser facilement, de manière à permettre la réalisation de cette tâche sans délai et le mieux possible.Or, dans notre société moderne, il est rare de laisser une profusion de moyens pour réaliser une tâche. Le manque de moyen est souvent générateur de vulnérabilité. Ce facteur ici généralise le concept de budget présenté dans [73].

Position : Ce facteur définit l'importance du poste occupé par l'opérateur dans l'organisation. Cette importance peut-être de plusieurs natures : un poste peut être important par sa position hiérarchique, mais aussi par sa criticité dans le système. Par exemple, un opérateur ayant un rôle technique essentiel pour le bon fonctionnement de l'organisation pourra avoir une valeur élevée alors même que son supérieur hiérarchique pourra lui avoir une valeur peu importante. Ce facteur permet de prendre en compte en partie l'impact que peut avoir un individu sur son SoSTS.

5.1.3 Quantification des facteurs directs et indirects

Dans la section précédente, nous avons dressé la liste des facteurs directs et indirects à prendre en compte pour modéliser les opérateurs d'un SoSTS dans le but d'en évaluer la vulnérabilité humaine. Dans ce contexte, il faut être capable de les quantifier de manière à pouvoir effectuer des traitements en vue de déterminer cette vulnérabilité.

Un grand nombre de facteurs peuvent être facilement quantifiés numériquement. Il est en effet possible d'utiliser une échelle numérique pour cela. Dans notre contexte, nous avons estimé qu'une échelle allant de 1 à 5 était suffisante. Cette approche est relativement intuitive (le niveau 1 étant le niveau le plus faible, le niveau 5 étant le niveau le plus élevé). De manière à guider l'architecte dans son choix, le tableau 5.1 vient synthétiser l'échelle des valeurs que nous avons mises en place, en donnant à chacune des valeurs une définition qui peut être contextualisée en fonction du facteur. Naturellement, cette échelle pourrait être modifiée et adaptée lorsque le contexte le requiert. On pourrait imaginer une échelle de 1 à 10, voire plus. Cela ne génerait en rien notre approche qui est complètement paramétrable pour le prendre en compte.

Parmi les facteurs que nous avons identifiés, certains sont difficilement caractérisables par une échelle numérique. En effet, si la compétence ou l'expérience peuvent être quantifiées simplement sur cette échelle, la stabilité émotionnelle ou la coopération organisationnelle d'un opérateur sont plus difficiles à évaluer numériquement. De ce fait, nous avons

Valeur	Définition
1	C'est le niveau le plus faible possible pour un facteur. Cela représente l'absence quasi totale du facteur soit au niveau du profil requis pour le poste soit au niveau de la capacité réelle de l'individu.
2	Ce niveau représente une attente faible pour le facteur. Cela pourrait s'illustrer par un niveau d'exigence sur le facteur qui importe peu.
3	Ce niveau représente la moyenne pour un facteur. Cela illustrerait par exemple un niveau d'expertise attendu moyen ou un niveau d'exigence moyen pour le facteur.
4	Ce niveau représente une attente pour le facteur qui est élevée. Cela s'illustrerait par exemple par un niveau d'expertise attendu qui serait supérieure à la moyenne.
5	Ce niveau représente le plus haut niveau possible qui soit pour un facteur.

TABLE 5.1 – Définition de l'échelle de valeurs pour les facteurs numériques

fait le choix de permettre une évaluation moins dépendante du contexte lorsque cela nous apparaissait peu clair vis-à-vis du facteur à évaluer. Pour cela, nous sommes appuyés sur l'approche utilisée dans le *Computer Security Handbook* [67]. Cette approche a pour but de venir qualifier les différentes qualités des collaborateurs pour permettre une meilleure sécurité par le management. Nous avons donc repris ce principe en nous appuyant sur des adjectifs afin de définir les différentes valeurs présentes pour chacun des facteurs complexes à définir numériquement. Nous nous sommes appuyés sur plusieurs adjectifs déjà présents dans cet ouvrage pour définir les adjectifs des facteurs concernés. Dans certains cas où cela était peu opportun, nous nous sommes appuyés sur la littérature.

Pour les facteurs suivants, nous proposons une évaluation utilisant les adjectifs associés. Pour garder une conformité avec les diagrammes illustratifs, nous avons laissé les valeurs en anglais mais y avons associé une traduction française.

Conscience : `efficient (efficace)`, `responsible (responsable)`, `compulsive (compulsif)`, `pompous (pompeux)`, `slavish (servile)`. Pour définir ce facteur, nous avons repris les adjectifs de [67]. En effet, ils sont suffisants pour qualifier les différentes combinaisons possibles, cela sans ambiguïté. Le tableau 5.2 donne les définitions utilisées.

Communication : `active(active)`, `energetic (énergétique)`, `quiet (calme)`, `shy (discrète)`, `silent (silencieuse)`. Les adjectifs choisis ici [67] permettent de décrire les principaux comportements en matière de communication. Le ta-

Adjectif	Définition
`Efficient`	Individu dont la conscience au travail est de réaliser correctement sa tâche sans plus.
`Responsible`	Individu qui aura tendance à être volontaire dans la réalisation de ses tâches ainsi que dans la prise de responsabilité vis-à-vis de tâches pouvant émerger.
`Compulsive`	Individu qui aura tendance à être instable dans sa capacité à s'investir dans son travail.
`Pompous`	Individu qui dans la réalisation de ses tâches sera extrêmement rigoureux. Cela pourra même aller jusqu'à un manque de flexibilité pour prendre en compte des éléments imprévus.
`Slavish`	Individu qui aura tendance à appliquer toute nouvelle décision qui lui sera imposée et cela sans remise en question.

TABLE 5.2 – Définition des adjectifs pour le facteur `Conscience`

bleau 5.3 donne les définitions utilisées.

Adjectif	Définition
`Active`	Besoin de communication standard sans restriction.
`Energetic`	Besoin d'une communication beaucoup plus fournie. Le manque de celle-ci pourrait altérer la réalisation des tâches.
`Quiet`	Besoin d'une communication faiblement fournie. Trop d'information pourrait nuire à la réalisation de tâches.
`Shy`	Besoin d'une communication discrète. Une communication trop visible ou trop bruyante pourrait nuire à la réalisation des tâches.
`Silent`	Besoin d'une communication absente.

TABLE 5.3 – Définition des adjectifs pour le facteur `Communication`

Coopération organisationnelle : `Trusting (confiant)`, `Deferent (respectueux)`, `Reticent (réticent)`, `Untrusting (méfiant)` Les valeurs des adjectifs ont été extraits de travaux retraçant la capacité d'un individu à placer sa confiance ainsi que sa croyance dans un système technique pour permettre une bonne coopération [66]. Nous avons étendu cette notion aux individus et aux organisations avec laquelle le sujet a besoin de coopérer. Le tableau 5.4 donne les définitions utilisées.

Stabilité émotionnelle :`stable (stable)`, `unemotional (sans émotion)`, `anxious (anxieux)`, `moody (lunatique)` . Pour ce facteur nous avons réduit le nombre d'adjectifs présents dans [67], de manière à en simplifier la notation. Pour bâtir

Adjectif	Définition
`Trusting`	L'individu a confiance dans son organisation.
`Deferent`	L'individu adhère aux valeurs de son organisation.
`Reticent`	L'individu émet des réticences vis-à-vis de son organisation.
`Untrusting`	L'individu n'a pas confiance dans son organisation.

TABLE 5.4 – Définition des adjectifs pour le facteur `Coopération Organisationnelle`

cette simplification, nous sommes concentrées sur les éléments les plus distants, de manière à permettre d'établir une qualification sans ambiguïté pour l'architecte. Le tableau 5.5 donne les définitions utilisées.

Adjectif	Définition
`stable`	Individu stable émotionnellement.
`unemotional`	Individu qui est capable de réaliser son travail sans aucune émotion.
`anxious`	Individu qui est anxieux dans son environnement de travail.
`moody`	Individu qui est instable dans sa capacité à gérer son stress.

TABLE 5.5 – Définition des adjectifs pour le facteur `Stabilité émotionnelle`

Culture : `simpleStructure (Structure simple)`, `machineBureaucracy (machine bureaucratique)` , `professionnalBureaucracy (bureaucratie professionnelle)`, `divisionalisedForm (Forme divisé)`, `adhocracy (adhocratie)`. Les adjectifs retenus pour ce facteur ont été tirés du livre *Software of Mind* [74].Ils permettent de qualifier les grandes catégories de culture d'entreprise. Le tableau 5.6 donne les définitions utilisées.

Management : `organizational (organisationnel)`, `cognitive-Behavioral (cognitivo-comportementale)`, `relaxation (relaxation)`, `multimodal (multimodale)`, `individualFocus (focus individuel)`. Pour ce facteur, nous avons retenu l'approche issue de la méta-analyse [75] qui dresse un tableau des principales méthodes managériales et qui les classe dans de grandes catégories. Cette approche nous permet d'avoir des catégories distinctes et recouvrant l'ensemble des types de management. Le tableau 5.7 donne les définitions utilisées.

Adjectif	Définition
`SimpleStructure`	Désigne ici une structure simple avec un ordre hiérarchique pyramidal et avec une supervision directe.
`MachineBureaucratie`	Désigne ici une technostructure avec des mécanismes de coordinations qui vont permettre une normalisation des différents processus de travail.
`ProfessionnalBureaucracy`	Cet adjectif désigne ici une organisation avec un noyau opérationnel qui va permettre de répartir les différentes tâches en se focalisant sur la standardisation des compétences.
`DivisionalisedForm`	Cet adjectif désigne une organisation avec une ligne médiane permettant une répartition des tâches pour permettre une standardisation des résultats.
`Adhocracy`	Désigne une organisation avec des mécanismes d'ajustement mutuel permettant une coordination plus décentralisée.

TABLE 5.6 – Définition des adjectifs pour le facteur `culture`

Adjectif	Définition
`Organizational`	Cet adjectif désigne ici le management ayant pour but de supprimer les causes du stress dans les organisations en modifiant les pratiques des politiques organisationnelles comme celle qui concerne le leadership, le temps de travail, la santé et la sécurité au travail.
`Cognitive-Behavioral`	Cet adjectif désigne un management ayant pour but d'aider les individus à développer de nouvelles réponses comportementales face aux événements stressants. Pour cela ce management cherche à favoriser des stratégies émotionnelles axées sur les antécédents des individus pour permettre une meilleure compréhension du stress et des événements stressants.
`Relaxation`	Cet adjectif désigne ici un management ayant pour but de relaxer les individus pour diminuer le stress à travers différents exercices de relaxation.
`Multimodal`	Cet adjectif désigne ici un management qui combine le management organisationnel avec au moins un autre type de management.
`IndividualFocus`	Cet adjectif désigne ici un management ne rentrant pas dans les autres catégories et qui se focalisent sur un individu en lui fournissant des outils pour l'aider à gérer son stress au travail.

TABLE 5.7 – Définition des adjectifs pour le facteur `Management`

5.2 Définition du langage HoS-ML

Dans la section précédente, nous avons proposé un modèle conceptuel permettant de modéliser un humain dans un SoSTS à l'aide de facteurs décrivant les caractéristiques humaines à prendre en compte pour évaluer la vulnérabilité humaine. Dans cette section, nous proposons un langage nommé HoS-ML (*Human-Oriented Security architecture Modeling Language*). Celui-ci permet de soutenir la modélisation de l'architecture d'un SoSTS présentée en amont. Pour accompagner ce langage, un logiciel a été développé et permet la manipulation de celui-ci. Dans un premier temps, nous décrivons l'approche conceptuelle que nous proposons afin de modéliser le SoSTS et ses opérateurs humains. Puis, dans un deuxième temps, nous détaillons le métamodèle de ce langage pour ensuite présenter la sémantique et la syntaxe de HoS-ML. Enfin, nous détaillons la réalisation de l'outil qui s'appuie sur l'environnement Eclipse.

5.2.1 Approche conceptuelle

Afin d'évaluer la vulnérabilité humaine dans un SoSTS, il convient de permettre à un architecte de modéliser le SoSTS lui-même. Nous avons choisi de nous concentrer sur les sous-systèmes humains et leurs interactions. Pour cela, nous devons proposer à l'architecte un langage de description d'architecture socio-technique de système de systèmes. Les éléments à prendre en compte ont été décrits précédemment. Il s'agit de permettre la modélisation des humains ayant un rôle actif dans le SoSTS, et plus exactement des facteurs directs et indirects de chacun. En réalité, ce qu'on modélise pour cela, c'est moins l'opérateur physique qui occupera un poste mais les caractéristiques humaines nécessaires à l'occupation de ce poste. Nous appelons cela un *Rôle*. On doit également donner la possibilité de représenter les interactions entre les différents rôles ainsi définis, telles que le passage de d'informations (consignes, données, ...). On parlera plus généralement de *transmission de documents.* De tels systèmes reposent la plupart du temps sur des chaînes hiérarchiques. On doit donc pouvoir représenter également la notion de responsabilité entre un chef assumant une responsabilité et la personne qui, va accomplir une tâche relevant de cette responsabilité. Nous parlons dans la suite de *délégation d'objectif.* Enfin, dans le but de mettre en place les bases nécessaires à l'évaluation de la vulnérabilité que nous traiterons au chapitre suivant, il convient de permettre à l'architecte de représenter un opérateur humain occupant un poste ainsi défini. Là encore, on ne modélise pas par essence une personne physique mais les caractéristiques nécessaires à la représentation

d'une personne physique occupant potentiellement un poste. On parlera d'*Acteur* jouant un rôle.

5.2.2 HoS-ML : un langage inspiré de STS-ML

Afin de permettre la conception d'un SoSTS, nous proposons un langage appelé HoS-ML. Ce langage s'inspire du langage STS-ML [76] présenté dans l'état de l'art. L'objectif de STS-ML est de permettre la création d'un recueil d'exigences de sécurité pour la conception d'un système socio-technique. Pour cela, il représente les systèmes socio-techniques à travers plusieurs vues. La première vue est la vue architecturale dite *social view*. Dans cette vue, la représentation de l'architecture du système socio-technique est faite à travers la représentation des Rôles et des Acteurs, complétés de leurs différents objectifs et documents. Dans cette vue, les acteurs et rôles sont définis à travers la possession d'objectifs et de documents et n'ont pas de définition en soi en dehors de leur nom. La deuxième vue de ce langage est *information view*. Elle est centrée sur la modélisation des informations présentes dans l'architecture et plus précisément elle identifie quel rôle détient quelles informations. La troisième vue, *authorization view*, a pour but de définir les autorisations de chacun vis-à-vis de différentes informations et objectifs définis dans l'architecture.

Pour créer HoS-ML, nous avons fait le choix de nous inspirer de STS-ML pour proposer un langage permettant de représenter un SoSTS intégrant le modèle de facteur humain que nous avons présenté en amont. En effet, la manière dont STS-ML représente une architecture socio-technique à l'aide de rôles et d'acteurs dans son diagramme *social view* est particulièrement adaptée à nos besoins. L'adaptation d'une représentation d'un système socio-technique à un système de systèmes socio-technique ne pose pas de problème en soit. En effet, dans notre approche, nous considérons que l'humain en tant que tel peut-être lui-même vu comme un système.

Pour définir HoS-ML, nous avons donc réutilisés des éléments de la *social view* du métamodèle de STS-ML, et nous y avons ajouté les concepts permettant la prise en compte de la vulnérabilité humaine. La figure 5.2 représente le métamodèle que nous proposons. Les éléments inspirés de STS-ML sont en bleu. Ils permettent la représentation de l'architecture d'un SoSTS. Les éléments que nous ajoutons sont en rouge. Nous détaillons le métamodèle à la section suivante.

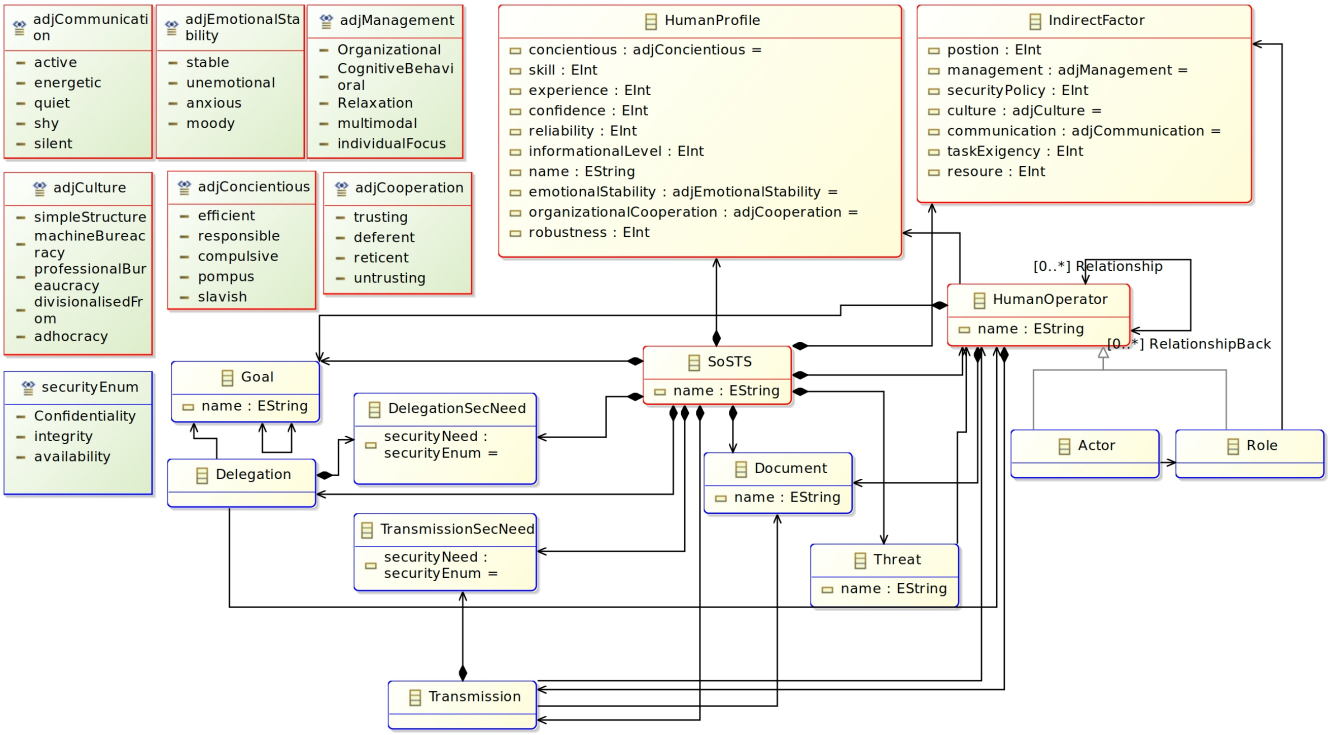


FIGURE 5.2 – Métamodèle de HoS-ML

5.2.3 Sémantique et syntaxes du langage

Nous donnons ici la description du langage HoS-ML. Pour cela, dans un premier temps nous donnons la syntaxe abstraite de la sémantique du langage et dans un second temps la syntaxe concrète que nous avons définie. Nous illustrons l'utilisation du langage à travers la modélisation d'une architecture d'un SoSTS simpliste.

5.2.3.1 Syntaxe abstraite

Le métamodèle visible la figure 5.2 permet de représenter les différents éléments composant la syntaxe abstraite du langage HoS-ML. Ici nous décrivons chaque élément composant ce métamodèle et expliquons leur rôle dans le langage :

`SoSTS` cette métaclasse désigne l'architecture finale à modéliser. Celle-ci est composée de tous les autres éléments présents dans le métamodèle.

`HumanOperator` cette métaclasse est la représentation abstraite d'un individu dans un SoSTS. Cette métaclasse n'est pas directement utilisée dans la représentation d'une architecture puisqu'elle est spécialisée dans les deux métaclasses sui-

vantes : `Role` et `Actor`. Afin de représenter cet acteur dans architecture, la métaclasse est liée à la métaclasse `HumanProfil` permettant de représenter les différents facteurs directs de cet acteur. Cette métaclasse est également liée aux métaclasses `Delegation` et `Goal` pour représenter les objectifs liés à un acteur et la délégation de ces objectifs de/vers un autre acteur. Un lien existe aussi vers les métaclasses `Transmission` et `Document` afin de représenter d'une part la possession d'une information et d'autre part sa transmission de/vers un autre Acteur. La boucle `Relationship` est ce qui permet de crée des liens relationnels entre les opérateurs humains dans l'architecture. Les relations sont des paramètres appartenant directement à l'élément concerné et n'apparaissent pas dans l'architecture en tant que lien entre les opérateurs humains.

`Role` cette métaclasse représente le rôle qu'un acteur va devoir occuper dans un SoSTS. C'est en fait une description du profil humain attendu sur un poste donné. Ce rôle va notamment avoir un lien supplémentaire à celui de la métaclasse `Human-Operator` : il s'agit du lien vers les facteurs indirects permettant de définir l'environnement de ce rôle.

`Actor` Cette métaclasse représente ici l'individu qui va occuper dans l'architecture un ou plusieurs rôles.

`HumanProfil` Cette métaclasse représente tous les facteurs directs qui ont été définis dans le modèle vulnérabilité humaine précédemment défini. Ils vont venir caractériser le profil idéal attendu pour un rôle et le profil effectif d'un acteur jouant le rôle. Les énumérations présentes sur la gauche du métamodèle représentent les différentes valeurs possibles pour les facteurs quantifiés avec un adjectif.

`IndirectFactor` Cette métaclasse représente les facteurs indirects qui vont venir décrire l'environnement dans lequel évoluent les rôles.

`Goal` Cette métaclasse représente les objectifs qui vont être utilisés, soit dans les rôles, soit par les acteurs considérés dans l'architecture.

`Document` Cette métaclasse représente les documents et informations qui sont attachés à un poste décrit par un rôle et qui sont donc en possession des différents acteurs assurant ces rôles.

`Delegation` Cette métaclasse représente la délégation qu'il peut y avoir lorsque plusieurs `Actor` travaillent ensemble autour d'un même objectif. Cela permet de représenter les liens structurels dans l'architecture entre les différents acteurs.

`Transmission` Cette métaclasse représente la transmission de documents et d'informations qu'il peut y avoir entre plusieurs `HumanOperator`. Ce lien permet de représenter la transmission de documents ou d'informations dans l'architecture.

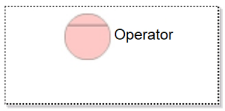
`DelegationSecNeed` Cette métaclasse représente le besoin de sécurité nécessaire lorsqu'il y a usage d'une délégation pour représenter une collaboration autour d'un même objectif de la part de plusieurs acteurs.

`TransmissionSecNeed` Cette métaclasse représente le besoin de sécurité nécessaire lorsqu'il y a transmission de documents ou d'informations entre plusieurs acteurs de l'architecture.

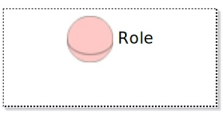
`Threat` Cette métaclasse représente une possible menace qui va cibler un acteur dans l'architecture. Cette menace permet de désigner la cible potentielle d'une attaque. Cette information sera nécessaire au processus d'évaluation de la vulnérabilité que nous présenterons au chapitre suivant.

5.2.3.2 Syntaxe concrète

Après avoir défini la syntaxe abstraite, il convient de définir une syntaxe concrète permettant de représenter les éléments de syntaxe abstraite en vue de leur manipulation dans un langage. Nous définissons une syntaxe concrète graphique de manière à représenter les modèles des architectures à modéliser de manière simple. En effet, nous croyons que les représentations graphiques auxquelles sont habitués les architectes via des langages tels que UML et SysML sont maintenant largement admises et utilisées. Ces représentations nous semblent plus simples à manipuler que des langages textuels. Pour ce faire, nous présentons les différents éléments composant le langage ci-dessous :

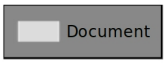


Cet élément représente l'acteur dans l'architecture. Les éléments modifiables pour cet acteur sont le nom ainsi que les différents facteurs directs qui le caractérisent. De plus, le carré blanc autour de l'acteur est là pour contenir les différents documents et objectifs lui appartenant.



Cet élément permet la représentation d'un rôle dans l'architec-

ture. Les paramètres modifiables pour le rôle sont le nom ainsi que le profil humain idéal. Celui-ci est représenté par les facteurs directs et les facteurs indirects qui y sont liés. De la même manière que pour l'acteur, le carré blanc autour du rôle est là pour contenir les différents éléments lui appartenant, à savoir les différents documents et objectifs.



Cet élément est la représentation d'un document ou d'une information dans l'architecture. Le document peut appartenir soit à un rôle, soit un acteur. Il peut être nommé de manière libre. Il peut être transmis à un autre acteur ou un autre rôle grâce à un élément transmission.

Cet élément permet la représentation d'un objectif dans l'architecture. L'objectif peut appartenir soit à un rôle, soit à un acteur. On peut modifier le nom de l'objectif pour permettre son identification. Un objectif peut être délégué à un autre acteur ou à un autre rôle grâce à l'élément délégation.

Cet élément permet la représentation de la transmission d'une information ou d'un document dans l'architecture. La transmission va désigner d'un côté le document à transmettre et de l'autre l'acteur ou bien le rôle qui le reçoit. La transmission peut avoir trois types de contraintes de sécurité qui peuvent lui être ajoutées de manière indépendante. Ces trois contraintes sont la confidentialité, l'intégrité et la disponibilité, qui sont des contraintes usuelles en termes de sécurité.

Cet élément est la représentation de la délégation d'un objectif dans l'architecture. La délégation est d'un côté liée à un objectif et va de l'autre côté être liée à l'acteur ou au rôle qui va prendre part à la réalisation de l'objectif. La délégation d'objectifs peut avoir les trois même types de contraintes de sécurité que la transmission. Elles peuvent lui être ajoutées de manière indépendante.

Cet élément permet la représentation de la menace dans l'architecture. La menace va être liée à un acteur ou à un rôle et va décrire une attaque sur l'opérateur humain sélectionné.

5.2.3.3 Utilisation de HoS-ML : un exemple

De manière à illustrer l'usage de la syntaxe concrète de HoS-ML, nous proposons le scénario suivant et sa modélisation avec HoS-ML.

Dans ce scénario, *Alice* joue le rôle d'une *infirmière*. Le rôle d'une infirmière a deux éléments spécifiques : le *dossier médical du patient* ainsi qu'un objectif qui est de *sauver la vie du patient*. *Bob* est un deuxième acteur. Il joue le rôle d'un *technicien de laboratoire*. Il a également deux éléments qui sont spécifiques à son rôle, qui sont les *résultats d'une analyse de sang* et l'objectif *d'analyser le sang*. L'infirmière participe à la réalisation de l'objectif *effectuer une analyse de sang* du *technicien de laboratoire*. Cette délégation d'objectif doit garantir la confidentialité. Le document contenant les résultats de l'analyse de sang doit être retourné à l'*infirmière* par le *technicien de laboratoire*. Là encore, la confidentialité est de mise. De plus, le document doit être transmis sans altérer ses données, c'est à dire que la transmission doit garantir également l'intégrité du document.

La figure 5.3 présente le modèle décrivant ce scénario. Par soucis de lisibilité, tous les facteurs n'ont pas été renseignés.

Dans ce scénario, le rôle `Nurse` est décrit comme devant avoir les facteurs directs suivants : `Compétence` de niveau 5, `Stabilité émotionnelle` à la valeur `stable`, et `Confiance` de niveau 4. Les facteurs indirects associés à ce rôle sont : `Management` à la valeur `CB` (acronyme pour `Cognitive-Behavioral`), `Resource` de niveau 4 et `Position` de niveau 4 également. Le rôle a sous sa responsabilité un document qui est le dossier médical du malade (`Medical Files`) et un objectif qui est `Save the patient's life`.

Le rôle `Lab Technician` a pour sa part les facteurs directs suivants : `Compétence` de niveau 3, `Stabilité émotionnelle` à la valeur `stable`, et `Confiance` de niveau 3. Les facteurs indirects associés à ce rôle sont : `Management` à la valeur `CB`, `Resource` de niveau 3 et `Position` de niveau 3 également. Le rôle a sous sa responsabilité un objectif qui est de pratiquer l'analyse de sang `analysing blood` et un document qui est le résultat de l'analyse de sang (`results of blood analysis`).

Deux relations sont ajoutées à ces rôles. Il s'agit des éléments en bleu et en orange qui sont respectivement la délégation d'objectifs et la transmission de documents. La délégation d'objectifs représente le fait que le rôle *Nurse* participe à l'un des objectifs (*analysing blood*) du rôle *Lab Technician*. Cette délégation d'objectifs a une contrainte de sécurité qui est la *confidentialité*. La transmission de documents, dans ce cas, correspond à la transmission des résultats d'analyses sanguines du technicien de laboratoire à l'infirmière. Ceci doit être réalisé avec des contraintes de sécurité liées à la confidentialité et à l'intégrité.

Dans cet exemple, on peut voir `Alice` et `Bob` qui sont respectivement les acteurs qui jouent le rôle de `Nurse` et de `Lab Technician`. Ils sont modélisés tous deux avec les mêmes facteurs que les rôles qu'ils occupent.

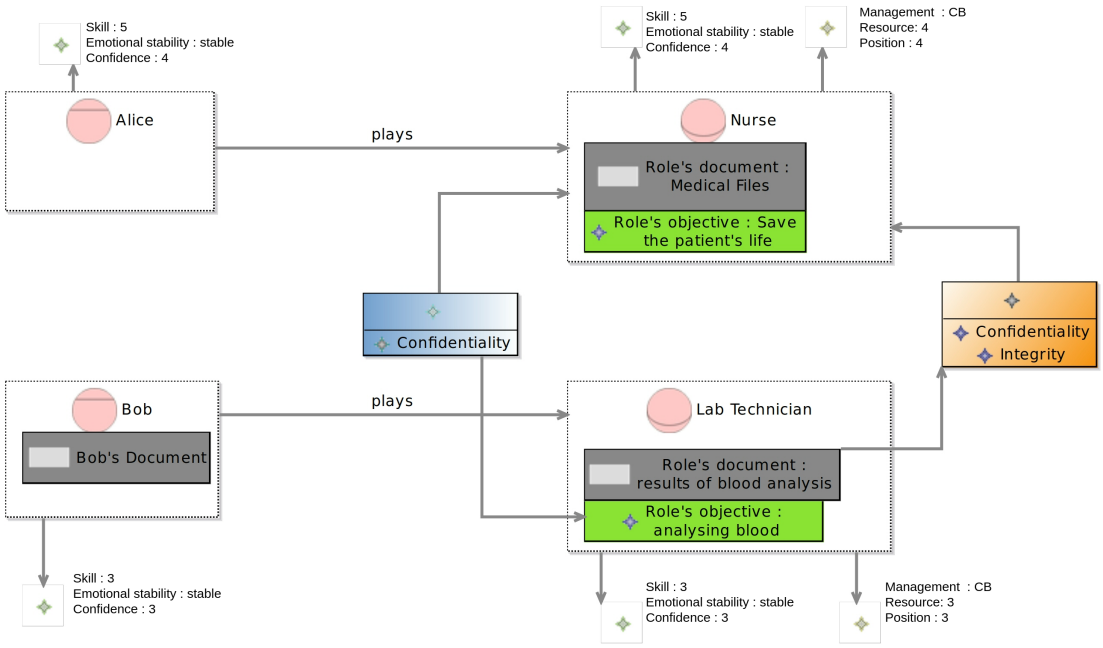


FIGURE 5.3 – Un exemple d'architecture de SoSTS utilisant HoS-ML

5.3 HoS-ML Editor : une implémentation du langage

Le langage HoS-ML étant un langage d'architecture de SoSTS, il faut permettre sa manipulation de manière simple et didactique. Pour ce faire, nous avons réalisé un logiciel implémentant le langage appelé HoS-ML Editor. Le logiciel réalisé est un logiciel à but démonstratif. Il se veut une preuve de concept. Aussi, son aspect est relativement brut et il n'a pas disposé d'une phase d'amélioration ergonomique dont devrait disposer un logiciel professionnel. Nous invitons le lecteur à considérer ce logiciel comme un prototype de recherche qui doit permettre la manipulation du langage et la démonstration de son fonctionnement.

Ce logiciel a été réalisé en utilisant la chaîne logicielle Sirius [1] qui est basée sur EMF et GMF.

1. Sirius : `https://www.obeosoft.com/fr/produits/eclipse-sirius`

5.3.1 Présentation de Sirius

Sirius est un progiciel qui a été créé par l'éditeur Obéo [2]. L'objectif de Sirius est de permettre la génération d'un logiciel permettant de manipuler un langage de modélisation. Sa particularité est de permettre au développeur d'effectuer sa génération de manière guidée en intégrant les frameworks que sont EMF et GMF. Ces derniers ont chacun un objectif spécifique dans la création du logiciel permettant la manipulation du langage de modélisation :

EMF (pour *Eclipse Modeling Framework*) est un framework qui prend en entrée le métamodèle du langage de modélisation que l'on souhaite utiliser dans un logiciel à générer et le transforme directement en code source pour permettre sa manipulation par le framework suivant.

GMF (pour *Graphical Modeling Framework*) est un framework qui prend le code généré par EMF et lui associe une syntaxe concrète d'une part, et des fonctions supplémentaires n'apparaissant pas dans le métamodèle d'autre part.

Nous avons synthétisé le processus de création d'un logiciel manipulant un langage de modélisation dans la figure 5.4. On peut distinguer le fait que chaque étape va utiliser un environnement qui lui est propre permettant au final d'avoir le logiciel contenant le langage de modélisation ainsi que ses fonctions.

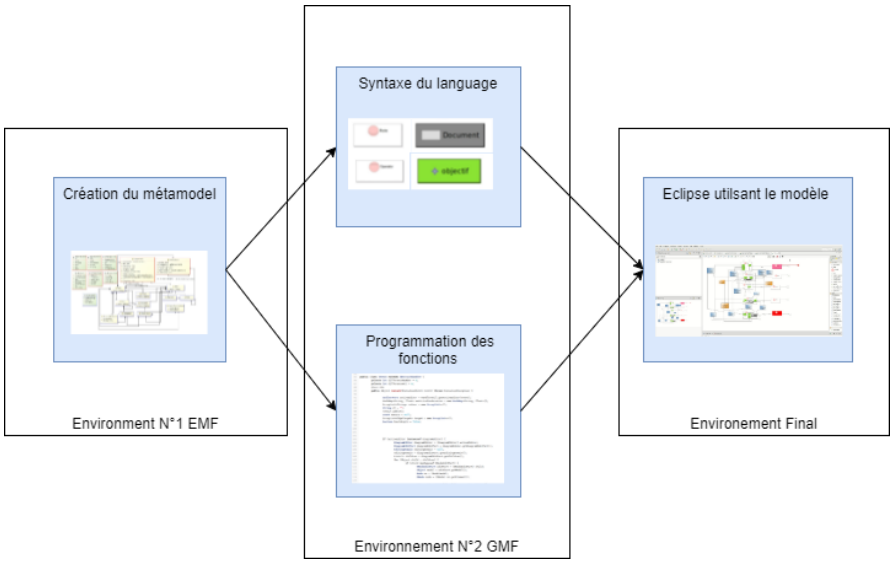


FIGURE 5.4 – Représentation du processus de création d'un logiciel avec Sirius

2. OBEO - `https://www.obeosoft.com/fr/`

5.3.2 Réalisation de HoS-ML Editor

Pour la réalisation de HoS-ML Editor, nous avons utilisé le métamodèle déjà présenté dans la figure 5.1. Une fois le métamodèle intégré dans Sirius, nous avons pu générer le code de celui-ci. Ce code est ensuite utilisé pour créer la syntaxe concrète de notre langage. Comme syntaxe concrète, nous avons utilisé celle présentée en amont. Nous avons ensuite ajouté des fonctionnalités supplémentaires. Par exemple, nous avons défini une fonction permettant de saisir le profil humain des rôles et des acteurs ainsi qu'une autre permettant de saisir les facteurs indirects des rôles. La figure 5.5 montre l'interface permettant la saisie des profils humains. Elle a été réalisée grâce à la personnalisation des fonctionnalités permises dans GMF.

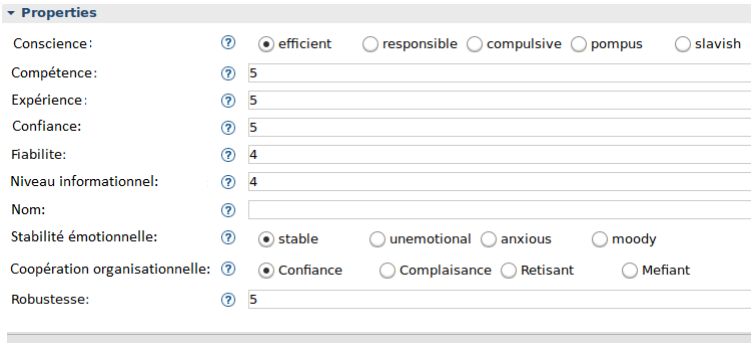


FIGURE 5.5 – Interface de paramétrage du profil humain

La figure 5.6 montre l'interface finale de HoS-ML Editor. Au centre de cette interface se trouve l'espace permettant de représenter l'architecture grâce aux différents éléments du langage HoS-ML. Sur la droite se trouve la palette permettant de sélectionner les éléments que l'architecte peut intégrer dans l'architecture. En bas se trouve l'interface de personnalisation permettant de modifier les attributs (le nom d'un élément de langage lorsqu'il en a un, le profil humain, ...) d'un élément de l'architecture sélectionné. Sur la gauche se trouve la gestion du projet qui va regrouper les différentes vues du système qui sont représentées avec le langage et forment ainsi le modèle complet de l'architecture du SoSTS étudié.

5.4 Résumé du chapitre

Dans ce chapitre, nous avons proposé notre vision de la modélisation de l'architecture d'un SoSTS de manière à en permettre l'étude sous le prisme de la détection de

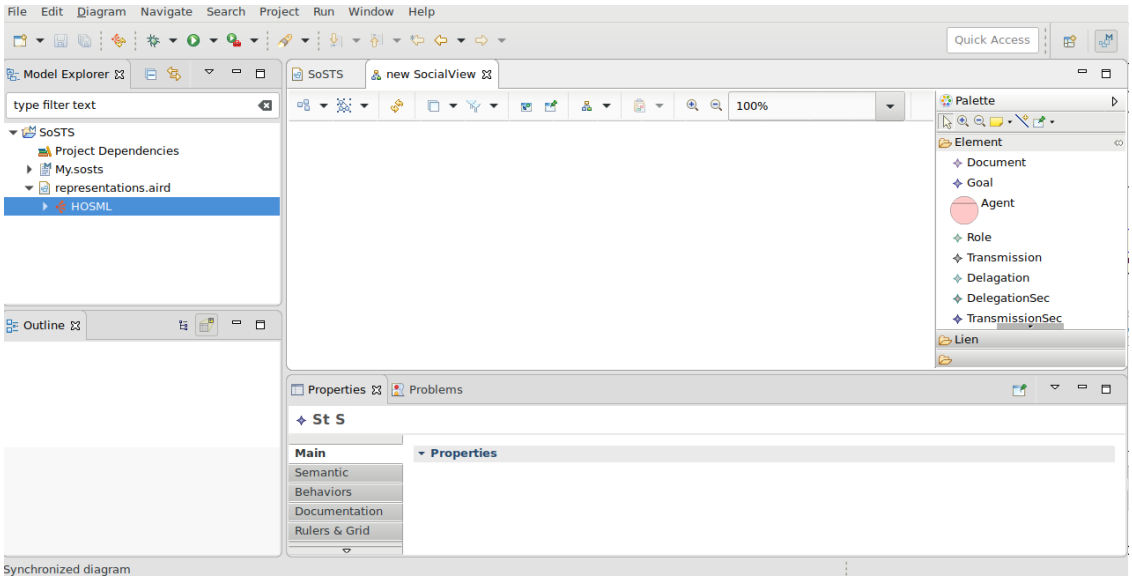


FIGURE 5.6 – Interface finale de HoS-ML Editor

la vulnérabilité humaine. Cette vision repose sur plusieurs contributions que nous avons détaillées.

La première est une proposition de modèle conceptuel de l'humain qui s'appuie sur deux types de caractéristiques que l'on appelle les facteurs humains. Il dit en substance qu'un humain peut-être modélisé par un ensemble de facteurs directs (qui représentent les propriétés de l'humain), de facteurs indirects (qui représentent l'environnement dans lequel il évolue) et de relations avec d'autres humains. Nous avons précisé ce modèle en constituant une liste de facteurs directs et indirects que nous avons identifiés comme étant significatifs dans l'évaluation de la vulnérabilité d'un humain. Nous avons proposé de quantifier ces facteurs via une échelle numérique lorsque c'était possible, ou avec des adjectifs quand cela permettait une meilleure caractérisation.

La deuxième contribution que nous avons développée ici a été le développement d'un langage de modélisation intégrant le modèle conceptuel proposé. Pour cela, nous avons défini un métamodèle spécifiant le modèle conceptuel proposé, puis nous avons implémenté celui-ci sous la forme d'un langage de modélisation appelé HoS-ML que nous avons outillé avec Sirius afin de créer un éditeur implémentant le langage appelé HoS-ML Editor.

Cette contribution vise donc à permettre à un architecte de réaliser des modèles de SoSTS embarquant les propriétés nécessaires à l'évaluation de la vulnérabilité humaine au sein de ces modèles. Dans le chapitre suivant, nous allons voir comment utiliser cette contribution pour évaluer la vulnérabilité des modèles réalisés.

Chapitre 6

Estimation de la vulnérabilité humaine dans un SoSTS

Sommaire

Dans le chapitre précédent, nous avons présenté notre première contribution. Celle-ci a permis le développement d'un langage de modélisation permettant à un architecte de modéliser l'architecture d'un SoSTS. Cette modélisation embarque un certain nombre de propriétés que nous allons utiliser dans ce chapitre afin d'évaluer la vulnérabilité humaine des modèles d'architecture réalisés. En effet, la deuxième contribution que nous apportons dans ce travail consiste en la définition d'un moyen d'évaluation de la vulnérabilité humaine en fonction du modèle de SoSTS réalisé à l'aide du langage HoS-ML. Pour cela, dans un premier temps, nous proposons un processus guidant l'architecte dans l'utilisation du langage HoS-ML et de son éditeur. Puis nous proposons d'utiliser une méthode probabiliste qui permet à partir de données existantes dans la littérature, d'estimer la

vulnérabilité d'un opérateur humain. Cette méthode, basée sur l'utilisation d'un réseau bayésien, est présenté dans la Section 6.2. De plus, à la manière dont on étudierait la possibilité d'une attaque physique à se propager dans un réseau, une fois la probabilité d'une vulnérabilité humaine identifiée, il convient de s'intéresser à la manière dont celle-ci peut se propager dans l'architecture spécifiée. Pour cela, nous proposons dans la Section 6.3 une approche paramétrable de calcul de l'impact possible de la propagation de cette vulnérabilité. Celle-ci utilise dans un premier temps un modèle *ad-hoc* de propagation, puis deux modèles de contaminations sociales. Enfin, dans la Section 6.4, nous montrons la manière dont nous avons intégrés les travaux de cette section à l'outil défini dans le chapitre précédent. Le travail de recherche qui est présenté ici fait l'objet de plusieurs publications [58], [77].

6.1 Processus d'estimation de la vulnérabilité humaine

Afin d'évaluer la vulnérabilité humaine dans un SoSTS, il nous faut dans un premier temps définir le processus d'usage du langage HoS-ML et de l'outil HoS-ML Editor. L'objectif est de guider l'architecte dans la conception des architecture des SoSTS, cela à travers un processus dédié et intégré utilisant le langage et l'outil proposés précemment.

Le processus que nous avons défini est basé sur 4 étapes, comme le montre la figure 6.1. Ces quatre étapes sont définies ci-dessous :

1 La première étape consiste, pour l'architecte système qui a la connaissance du SoSTS, à définir les différents rôles intervenant dans le système modélisé, en relation avec les éléments qui les composent : les objectifs et les documents. Un rôle est la représentation de l'opérateur idéal en termes de facteurs humains. L'architecte doit donc définir les différents facteurs directs idéaux pour ce rôle, et également définir les différents facteurs indirects qui permettent de définir l'environnement dans lequel va évoluer ce rôle. Ensuite, l'ingénieur système va ajouter les liens entre les différents rôles composant ce SoSTS afin de définir son architecture structurelle. Ces liens vont être les liens de transmission de documents et de délégation d'objectifs. L'ingénieur va pour finir ajouter les différentes relations humaines qui sont liées au rôle et qui vont permettre de décrire l'architecture humaine de ce SoSTS.

2 La deuxième étape consiste en une pré-analyse de l'architecture ainsi composée. Cette analyse vise à détecter les premières divergences entre facteurs directs et indirects liés au rôle, afin de vérifier qu'aucune vulnérabilité n'existe *de facto* dans

l'architecture. Il s'agit d'un premier niveau de détection des vulnérabilités pour vérifier avant tout la cohérence de cette architecture.

3 La troisième étape est celle où l'architecte va attribuer pour chaque rôle un acteur attaché à celui-ci. Ces derniers ne sont pas de réels humains. En effet, l'architecte ne peut pas connaître l'opérateur réel qui occupera au final sous rôle dans le SoSTS. De plus l'opérateur peut évoluer au fur et à mesure du temps voire peut changer au fur et à mesure de la vie du SoSTS. Cette affectation d'acteurs sert avant tout à simuler le cas où l'individu occupant réellement le poste ne présenterait pas tous les facteurs directs idéaux demandés par le rôle. Cette affectation permet la simulation de la vulnérabilité humaine et le calcul de l'impact potentiel d'une vulnérabilité sur ce rôle pour le système. L'objectif de cette étape est donc d'attribuer les différents acteurs aux rôles et de définir leurs facteurs directs et leurs relations humaines afin de permettre une première approche de simulation sur l'architecture.

4 La quatrième étape consiste en la simulation de la vulnérabilité humaine sur les acteurs. Dans cette étape l'architecte doit venir simuler la vulnérabilité des différents acteurs prenant place dans les différents rôles de l'architecture. Pour ce faire, il doit réaliser un scénario d'attaque. Puis, il va appliquer, en fonction des scénarios retenus, une menace correspondante sur l'acteur ciblé. La vulnérabilité humaine variant en fonction du profil de l'acteur, cette étape va de pair avec l'étape numéro trois. L'architecte va en effet faire plusieurs fois cette étape de simulation en modifiant le profil de l'individu. Il doit donc repasser par l'étape numéro trois pour refaire le profil des acteurs. Une fois ces différentes opérations réalisées, l'architecte va pouvoir avoir une vision étendue des différentes vulnérabilités humaines présentes dans l'architecture et qui correspondent à un scénario pouvant exploiter cette vulnérabilité.

Dans la suite de ce chapitre, nous définissons les moyens permettant de simuler et d'évaluer la vulnérabilité humaine à partir du modèle réalisé à l'aide d'HoS-ML Editor.

6.2 Une approche d'estimation de la vulnérabilité humaine

Pour un ingénieur concevant un SoSTS, estimer la vulnérabilité humaine, suite à une attaque cyber, revient à s'interroger sur la possibilité qu'un opérateur humain, dont il est

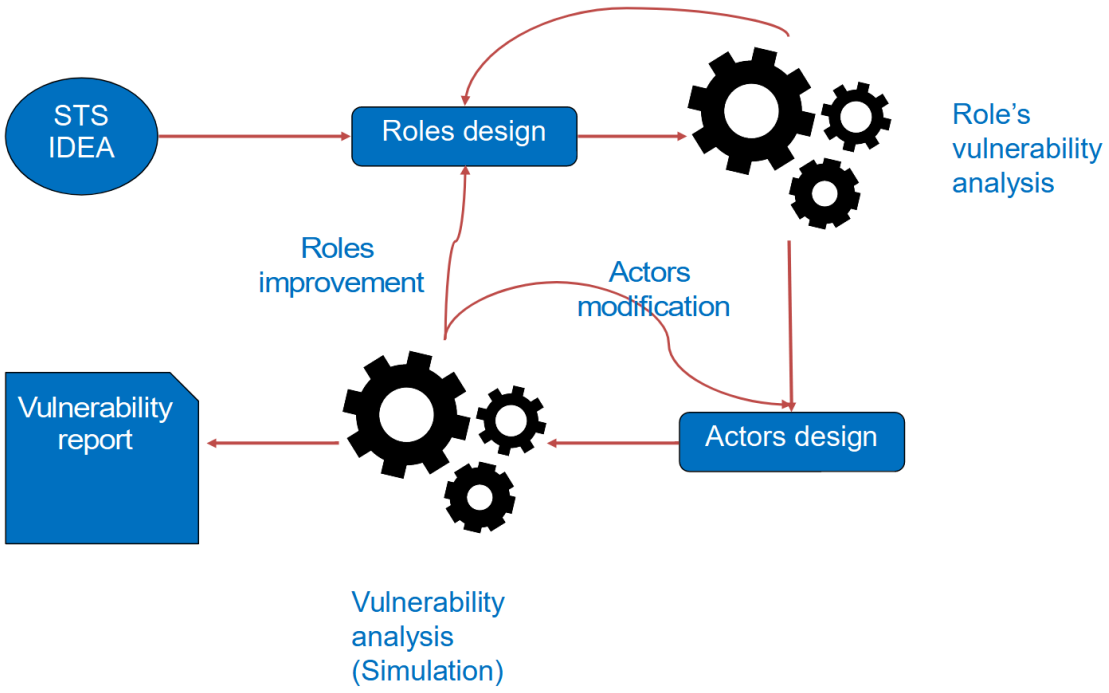


FIGURE 6.1 – Processus d'utilisation de HoS-ML

en train de concevoir le poste, puisse commettre une erreur volontaire ou involontaire. Évidemment, cela sans avoir une quelconque idée de la personne qui occupera réellement ce poste. Dès lors, il est aisé de comprendre la complexité d'une telle opération. Dans cette section, nous allons dans un premier temps présenter la méthode que nous proposons pour estimer une vulnérabilité humaine dans un SoSTS. Cette méthode s'appuie sur la définition du modèle humain présenté dans la section précédente. Dans un second temps, nous aborderons le moyen de calcul que nous avons défini. Celui-ci s'appuie sur une approche Baéysienne. Enfin, nous décrirons la manière dont nous utilisons les facteurs indirects pour enrichir l'estimation initiale de la vulnérabilité humaine.

6.2.1 Une méthode d'estimation la vulnérabilité humaine

Comme dans toute méthode d'ingénierie, définir un SoSTS passe par plusieurs étapes. Dans l'étape qui nous concerne, un ingénieur, après avoir identifié les sous-systèmes techniques nécessaires à une tâche opérationnelle, va identifier les différents opérateurs nécessaires au fonctionnement de cette même tâche. Notre étude étant centrée sur la vulnérabilité humaine, nous avons choisi un point de vue centré sur ces opérateurs. Pour l'ingénieur,

il s'agit d'identifier leurs *rôles* dans le système. L'étude de ces rôles va consister à identifier les exigences minimales en termes de facteurs directs que devront présenter les opérateurs physiques qui tiendront ces rôles. Pour ces opérateurs physiques, on parlera d'*acteurs*. Prenons par exemple le rôle boulanger à bord d'un navire . Étant donné son importance vis-à-vis du bien être de l'équipage, et les conditions dans lesquelles il va remplir son métier, on peut considérer qu'il devra être a minima expérimenté. Cela n'empêche pas un boulanger débutant d'occuper le poste. Cependant, ce delta entre l'expérience attendue décrite par le rôle, et l'expérience de l'acteur qui occupera effectivement le poste peut constituer une vulnérabilité.

Ainsi, nous proposons une méthode d'estimation de la vulnérabilité humaine utilisant cette différenciation entre les facteurs directs d'un rôle et ceux d'un acteur. Le rôle décrira les facteurs directs requis pour un poste donné. L'acteur décrira un opérateur humain occupant ce poste. En effet, lorsqu'un individu occupe dans la réalité un rôle, il ne correspond jamais parfaitement aux exigences minimales du rôle. Par moment, il sera bien au-dessus de tous les facteurs décrivant le rôle. Dans d'autres cas, l'acteur sera en décalage avec les facteurs identifiés par le rôle comme étant nécessaires à la réalisation des tâches qui lui incomberont. Dans ce contexte, nous posons l'hypothèse suivante ; lorsqu'un acteur ne possède pas tous les facteurs humains requis par le rôle, il peut représenter une vulnérabilité. En effet, lorsqu'un opérateur humain occupe un poste sans correspondre au profil type définit par un rôle, cela représente un risque qu'il ne soit pas capable d'occuper de manière optimale le poste. Ce risque ne représente pas une vulnérabilité avérée, mais simplement une probabilité que celle-ci existe et qu'elle puisse être exploitable.

Comme évoqué précédemment, les facteurs directs ne sont pas les seuls à pouvoir créer de la vulnérabilité. Les facteurs environnementaux, que nous avons appelés dans la section précédente facteurs indirects, ont également une influence sur la vulnérabilité. En effet, à titre d'exemple, un acteur étant soumis à un management mal adapté, risque de moins bien se comporter que prévu.

La méthode que nous proposons pour estimer la vulnérabilité des opérateurs humains peut donc être synthétisée de la manière suivante :

1. Pour un poste donné, la différence entre les valeurs des facteurs directs spécifiés pour un rôle, et celles des facteurs directs des acteurs remplissant ce rôle, peut augmenter la probabilité qu'un opérateur soit source de vulnérabilité pour le système.
2. De même, pour un poste donné, la différence entre les valeurs des facteurs indirects spécifiés pour un rôle, et celles des facteurs indirects des acteurs remplissant ce rôle,

peut renforcer une vulnérabilité éventuelle.

La figure 6.2 illustre cette approche en utilisant les facteurs directs et indirects que nous avons définis précédemment. A gauche, le rôle `first operator` est défini. Il est caractérisé par les couples `(facteurs directs - valeurs)` suivants : (`Experience - 3`), ce qui correspond à un individu ayant une expérience moyenne, ni débutant, ni un expert, par exemple un sous-officier en milieu de carrière ; (`Emotional stability - stable`), ce qui correspond à un individu gérant ses émotions lors de situations complexes ou tendues ; (`Reliability - 3`), ce qui traduit une capacité à ne généralement pas commettre d'erreurs ; (`Confidence - 2`), ce qui indique que l'individu qui occupera ce poste bénéficie d'une confiance plutôt faible de la part de son organisation dans sa capacité à respecter des règles de sécurité. Les facteurs indirects associés à la spécification de ce rôle sont les suivants ; (`Task exigency - 3`), ce qui indique que la tâche associée à ce rôle est d'un niveau d'exigence moyen ; (`Management - CB`) ce qui précise que le management est de type `CognitifBehavior`, c'est à dire que le management laisse les personnes en autonomie pour gérer une situation donnée ; (`Security Policy - 3`) ce qui précise qu'il est attendu du poste défini par le rôle un respect habituel des politique de sécurité ; (`Resource - 2`) ce qui indique que l'acteur occupant ce rôle aura plutôt peu de ressources annexes à mettre en action dans sa tâche. Sur la partie droite du diagramme, l'acteur `Alice` est définie. Alice n'est pas un acteur réel. C'est un acteur générique représentant un acteur occupant le rôle `First Operator`. Les valeurs des facteurs directs caractérisant `Alice` sont différentes de celles du rôle. Ainsi, elle a notamment une expérience plus faible, elle est d'une stabilité émotionnelle différente (elle est d'une nature anxieuse). Ce delta entre les valeurs des facteurs directs du rôle et de l'acteur peut être source de vulnérabilité. Nous verrons plus en aval comment le déterminer.

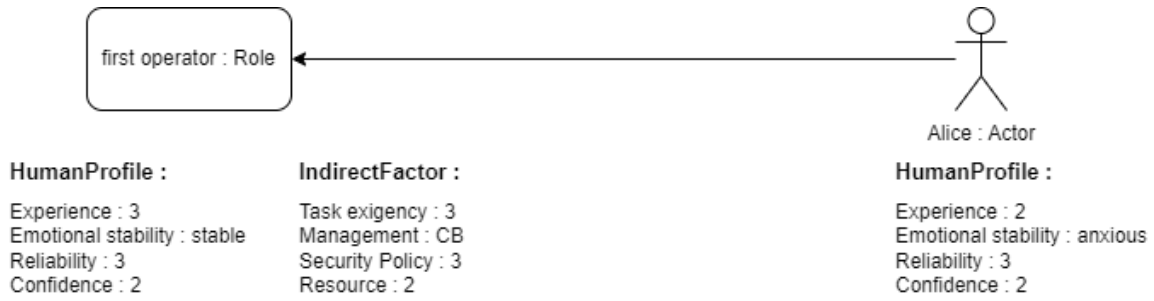


FIGURE 6.2 – Illustration de la méthode d'évaluation de la vulnérabilité humaine

Le résultat de la différence entre le profil minimal du rôle et le profil réel estimé de l'acteur est donc la base de calcul pour l'estimation de la vulnérabilité humaine. Ce delta qui sera calculé pour chaque facteur direct donnera une indication de l'écart entre le profil minimal attendu et le profil effectif. De plus les facteurs indirects auront un impact plus ou moins grand sur ce résultat. Dans la section suivante, nous proposons une approche mathématique permettant à la fois de donner un sens à cet écart et de retranscrire l'influence des facteurs indirects sur l'estimation de la vulnérabilité.

6.2.2 Détermination d'une méthode de calcul de la vulnérabilité humaine

Afin de déterminer une méthode de calcul adaptée, nous avons fixé les exigences suivantes. D'une part, la méthode de calcul doit permettre de combiner plusieurs facteurs séparés pour déduire une estimation de la vulnérabilité. D'autre part, la méthode de calcul doit pouvoir être paramétrée en fonction de données issues de la littérature ou de données propres au domaine industriel. Enfin, la méthode de calcul doit avoir des capacités d'apprentissage capables de prendre en compte un ou des contextes particuliers, en fonction par exemple des retours d'expérience de l'industriel. En effet, on peut imaginer une évolution du modèle en fonction de données nouvelles qui seraient construite sur l'expérience. Par exemple, imaginons un industriel développant un nouveau type de sous-marin avec un équipage réduit. Ses données initiales prennent en compte un type de sous-marin de génération plus ancienne, moins confortable mais à l'équipage plus nombreux. On peut aisément imaginer que la nouvelle configuration entraîne une évolution des données sociales que seule l'expérience pourra identifier.

Dans ce contexte, les approches utilisant un réseau bayésien sont les plus efficaces à notre connaissance. En effet, elles permettent de mettre en relation les différents facteurs ayant une interaction à prendre en compte via des nœuds. Ainsi, via des tables de probabilités, elles permettent de calculer, pour chaque facteur lié à un nœud, la vulnérabilité que la combinaison de valeurs des facteurs concernés pourrait induire. D'autre part, il est possible d'utiliser ces nœud comme des étapes de calcul intermédiaires. Ainsi, un nœud a sa propre valeur qu'il peut à son tour combiner avec d'autres valeurs de nœuds avec lesquels il a une relation via de nouvelles tables de probabilités ad-hocs. Cela permet notamment de prendre en compte l'influence des facteurs indirects sur les facteurs directs. En effet, on peut par exemple avoir un nœud issu des valeurs de facteurs directs que l'on

peut mettre en relation sémantiquement. Si un nœud indirect a une relation avec ce résultat, on peut à son tour lui appliquer l'approche. Construire le réseau bayésien consiste donc à déterminer les facteurs interdépendants entre eux, d'abord entre facteurs directs, puis en intégrant les facteurs indirects. Enfin, le réseau bayésien ainsi constitué peut dans un premier temps utiliser des valeurs probabilistiques existantes (issues de la littérature ou du domaine industriel), puis apprendre en fonction de nouveaux paramètres, et cela de manière indépendante pour chaque nœud. Ce dernier point permet d'apporter une expertise métier sur certain facteur qui pourrait être nécessaire pour estimer au mieux la vulnérabilité humaine dans certain cas de figure.

6.2.3 Mise en œuvre du réseau bayésien

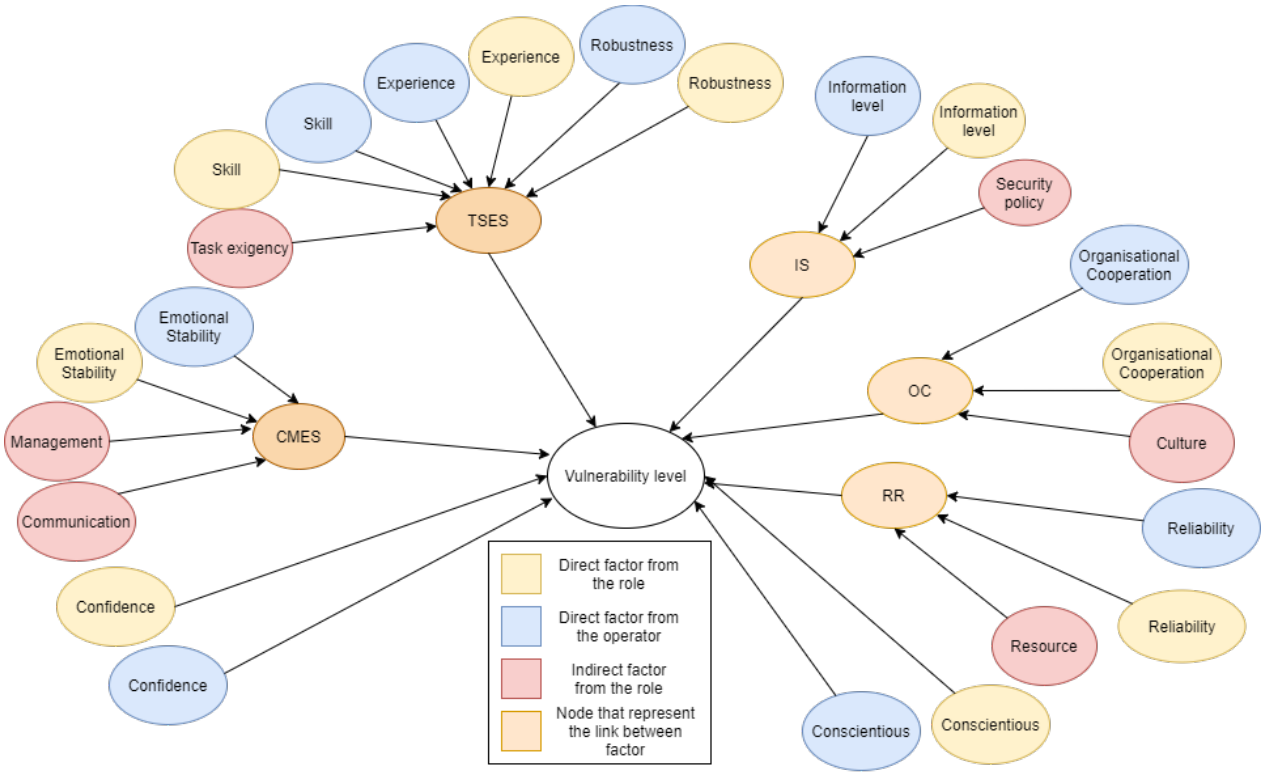
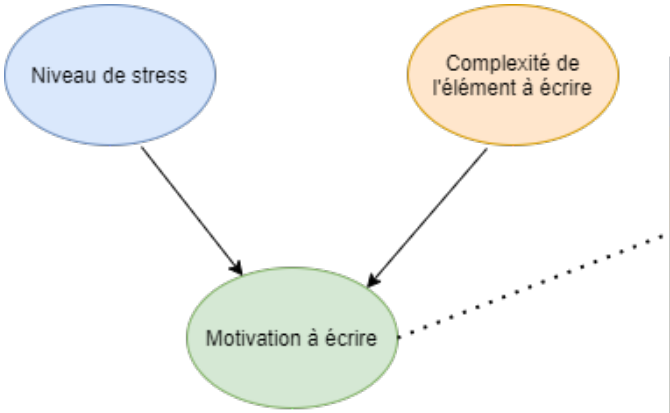


FIGURE 6.3 – Représentation du réseau bayésien évaluant la vulnérabilté humaine dans un SoSTS

6.2.3.1 Présentation du concept des réseaux bayésiens

Un réseau bayésien (RB) est un graphe orienté acyclique défini par [78] comme étant un modèle graphique probabiliste. Dans un réseau bayésien sont représentés par des nœuds des variables aléatoires composant un problème donné. Les liens représentent la dépendance probabiliste existant entre ces variables aléatoires. Cette dépendance probabiliste est ici une probabilité conditionnelle entre ces variables aléatoires. La figure 6.4 vient illustrer un exemple de réseau bayésien simple permettant d'illustrer son fonctionnement.

Les réseaux bayésiens sont souvent utilisés comme méthodes de machines learning [79]. En effet, ils permettent à un expert du domaine de venir renseigner ses propres données dans le réseau permettant ainsi au réseau de s'adapter à des contextes spécifiques.



Niveau de stress	faible(Fa)			moyen(M)			fort(Fo)		
Complexité de l'élément à écrire	Fa	M	Fo	Fa	M	Fo	Fa	M	Fo
Motivation à écrire : motivé	0.4	0.3	0.2	0.6	0.5	0.4	0.8	0.7	0.6
Motivation à écrire : non motivé	0.6	0.7	0.8	0.4	0.5	0.6	0.2	0.3	0.4

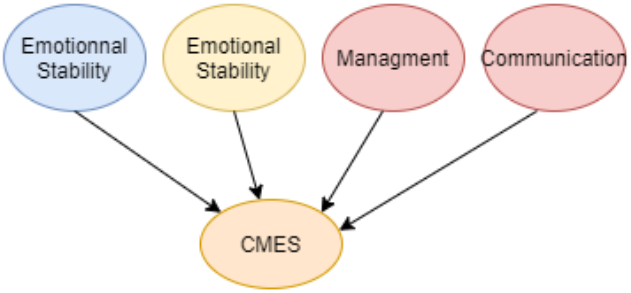
FIGURE 6.4 – Exemple de réseau bayésien

La figure 6.4 vient illustrer le scénario suivant : un opérateur humain doit écrire un livre. La motivation pour écrire ce livre dépend de deux choses : d'une part du niveau de stress pour l'écriture de celui-ci, d'autre part de la complexité de ce qu'il a à écrire. Les nœuds `Niveau de stress` et `Complexité de l'élément écrire` comptent trois états chacun : `faible`, `moyen` et `fort`. Le nœud `Motivation à écrire` a deux états : `motivé`, `non motivé`. Ainsi nous pouvons voir dans la table les probabilités du nœud *motivation à écrire* que plus le stress est important, plus l'individu sera enclin à écrire les éléments et cela même si la complexité freine grandement sa motivation. Par exemple, dans le cas où l'opérateur humain devrait écrire sous un fort stress un élément rédactionnel qui a une forte complexité, la probabilité sur le nœud *motivé a écrire* serait de : 0,6 pour motivé et 0,4 pour non motivé.

6.2.3.2 Le réseau bayésien

Nous détaillons ici la manière dont nous avons défini le réseau bayésien permettant de calculer la vulnérabilité humaine d'un SoSTS. La figure 6.3 illustre le réseau bayésien que nous avons défini. En jaune, sont représentés les facteurs directs du rôle ; en bleu, ceux de l'acteur ; en rouge, les facteurs indirects. Enfin, en orange, sont définis des nœuds intermédiaires pour lesquels une combinaison de facteurs directe et/ou indirects ont été identifiés. Ces derniers sont nommés via un acronyme constitué de la première lettre de chaque facteur impliqué. Afin de déterminer les relations existant entre les facteurs, nous nous sommes inspirés de la littérature, notamment issue des sciences humaines : [72], [75], [80]-[82].

Un nœud est caractérisé par son état. Celui-ci est fonction des probabilités correspondantes aux valeurs des facteurs impactant ce même nœud. Le nombre d'états pour chaque nœud peut-être important. Par exemple, le nœud `CMES` a 1875 états possibles. Il est donc impossible de donner l'ensemble du détail de chaque état d'un nœud, qu'il soit intermédiaire ou final. Pour illustrer le fonctionnement du calcul d'un état, la figure 6.5 montre une combinaison donnée du nœud `CMES` vis-à-vis des valeurs de ses sous-nœuds.



M	{Organizational, CognitiveBehavioral, Relaxation, Multimodal, IndividualFocus}
C	{active, energetic, quiet, shy, silent}
EM	{stable, unemotional, anxious, moody}
EM	{stable, unemotional, anxious, moody}
CMES	{bad, neutral, good}

FIGURE 6.5 – Exemple d'évaluation de la probabilité liée à une combinaison possible pour le nœud `CMES` dans le réseau bayésien.

6.2.3.3 Règles d'influence

Ci-dessous sont présentés les règles permettant de déterminer les états des différents nœuds intermédiaires :

IS : Selon [80], la politique de sécurité peut avoir un impact sur le facteur de niveau

informationnel. En effet, si celle-ci est suffisamment importante, elle peut venir compenser un manque de niveau informationnel. Nous avons transposé les éléments venant de [80] dans la règle 6.1. Le fait que la politique de sécurité doit être suffisamment importante a été traduit par un niveau minimal de celle-ci à quatre.

$$\boxed{SecurityPolicy \geqslant 4} \tag{6.1}$$

Cette règle donne donc comme résultat un nœud a trois états qui sont : **Bad** : cet état reflète le cas où l'individu ne remplit pas le minimum attendu et ce manque n'est pas compensé par l'influence de la politique de sécurité.
Neutral : Cet état reflète le cas où l'individu ne remplit pas le minimum attendu, mais ici le manque est compensé par l'influence de la politique de sécurité. Cet état peut aussi refléter le cas où l'individu remplit le minimum attendu, mais le minimum de sécurité lui n'apporte aucun bénéfice.
Good : Cet état reflète le cas où l'individu remplit le minimum attendu et la politique de sécurité vient ajouter un bénéfice. Cela diminuera le total du niveau de vulnérabilité de 1.

CMES : Le management et la communication ont un impact sur la stabilité émotionnelle d'après [75], [83]. Dans notre modèle nous avons transposé cette influence par le fait que, si le facteur `management` et le facteur `communication` sont en adéquation avec la valeur du facteur `stabilité émotionnelle` de l'acteur, cela diminue la vulnérabilité. La règle ainsi obtenue est la suivante :(6.2). Pour faire correspondre les différentes valeurs nous les avons extraits de [75], [83].

$$\boxed{\begin{array}{l}(Communication = active\ OR\ Communication = energetic)\ AND \\ \quad (\ (Management = CognitiveBehavioral\ AND \\ EmotionalStability(Actor) = Moody)\ OR \\ \quad\quad (Management = Multimodal\ AND\ EmotionalStability(Actor) = Stable)\ OR \\ \quad\quad (Management = IndividualFocus\ AND\ EmotionalStability(Actor) = Anxious)\ )\end{array}} \tag{6.2}$$

Cette règle donne donc comme résultat un nœud a trois états qui sont :

Bad : Cet état reflète le cas pour lequel l'individu ne correspond pas à l'état de stabilité émotionnelle attendue. Ce manque n'est d'ailleurs ici pas compensé par

l'influence du management.

Neutral : Cet état reflète le cas où l'individu ne correspondrait pas à l'état de stabilité émotionnelle, mais malgré cet écart, il y a une compensation amenée par l'influence du management. Cet état peut aussi refléter le cas dans lequel l'individu correspond à l'état de stabilité émotionnelle attendue, alors que le management n'apporte rien de plus.

Good : Cet état reflète le cas où l'individu correspondrait à l'état de stabilité émotionnelle attendue et le management apporte une influence positive. Cela diminuera le total du niveau de vulnérabilité de 1.

TSES : L'exigence de la tâche est une source de vulnérabilité si l'acteur ne dispose pas des compétences, de l'expérience et de la robustesse suffisantes. [72], [84]. Ainsi, nous considérons que le niveau de la vulnérabilité humaine n'augmentera pas si la somme des valeurs des trois facteurs `Compétence`, `Expérience`, et `Robustesse` est supérieure à trois fois la valeur du facteur `Exigence de la tâche`, comme défini par la règle (6.3) :

$$(Skill(Actor) + Experience(Actor) + Robustness(Actor)) \geqslant TaskExigency * 3 \tag{6.3}$$

Cette règle donne donc comme résultat un nœud a cinq états qui sont :

Bad4 : Cet état reflète le cas où l'acteur ne remplit pas le minimum attendu dans la compétence, l'expérience, la robustesse et l'influence de l'exigence de la tâche jouent en augmentant la vulnérabilité.

Bad3 : Cet état reflète le cas où l'acteur ne remplit pas le minimum attendu dans au moins deux des facteurs qui sont la compétence, l'expérience et la robustesse. Vient s'ajouter à cela l'influence de l'exigence de la tache qui va augmenter la vulnérabilité. Cela peu aussi refléter le cas pour lequel le minimum de l'acteur ne remplit pas le minimum attendu dans les trois facteurs, mais le total de ces facteurs est suffisant pour remplir la condition sur l'exigence de la tâche.

Bad2 : Cet état reflète le cas où l'acteur ne remplit pas le minimum attendu dans au moins un des facteurs qui sont la compétence, l'expérience et la robustesse. Vient s'ajouter à cela l'influence de l'exigence de la tâche qui va augmenter la vulnérabilité. Cela peu aussi refléter le cas pour lequel le minimum de l'acteur ne remplit pas le minimum attendu dans deux des facteurs, mais le total de ces

facteurs est suffisants pour remplir la condition sur l'exigence de la tâche.

Bad1 : Cet état reflète le cas où l'acteur ne remplit pas le minimum attendu dans au moins un des facteurs qui sont la compétence, l'expérience et la robustesse, mais le total de ces facteurs est suffisant pour remplir la condition sur l'exigence de la tâche.

Neutral : Cet état reflète le cas où l'acteur remplit le minimum attendu dans les différents facteurs que sont la compétence,l'expérience et la robustesse.

OC : ce nœud représente l'impact de la capacité de coopération organisationnelle de l'acteur sur sa vulnérabilité humaine. En effet, un acteur humain qui a des difficultés à coopérer sera plus vulnérable dans une entreprise où la coopération est nécessaire [81]. Ainsi, la vulnérabilité humaine augmentera dans le cas décrit par la règle (6.4). Pour transposer cette règle, nous avons pris les états venant décrire la défiance envers la coopération organisationnelle d'un individu et nous l'avons fait correspondre aux structures qui nécessitent de la coopération.

$$\begin{array}{l}(Organizationalcooperation(Actor) = Reticent\ OR \\ \quad Organizationalcooperation(Actor) = Untrusting)\ AND \\ (Organizationalcooperation(role) = simpleStructure\ OR \\ \quad Organizationalcooperation(role) = machineBureaucracy\ OR \\ \quad Organizationalcooperation(role) = professionnalBureaucracy)\end{array} \tag{6.4}$$

Cette règle donne donc comme résultat un nœud a trois états qui sont :

Bad2 : Cet état reflète le cas où l'individu ne correspond pas à l'état de coopération organisationnelle attendue. De plus, cet état entre dans ce qui peut augmenter la vulnérabilité en fonction de la culture d'entreprise.

Bad1 : Cet état reflète le cas où l'individu ne correspond pas à l'état de coopération organisationnelle attendue. De plus, cet état n'est pas en conflit avec la culture d'entreprise.

Neutral : Cet état reflète le cas où l'individu correspond à l'état de coopération organisationnelle attendue.

RR : ce nœud représente l'impact de la ressource sur la fiabilité. Un apport suffisant de ressources peut permettre de diminuer une vulnérabilité [82] . Ainsi nous considérant que la vulnérabilité humaine diminuera si le facteur `Ressource` est suffisant.

Ce que nous traduisons par la règle suivante(6.5) :

$$\boxed{Resource(Role) > 4} \tag{6.5}$$

`Resource > Reliability`.

Cette règle donne donc comme résultat un nœud a trois états qui sont :

Bad : Cet état reflète le cas où l'individu ne remplit pas le minimum attendu en fiabilité et ce manque n'est pas compensé par l'influence de la ressource .
Neutral : Cet état reflète le cas où l'individu ne remplit pas le minimum attendu en fiabilité, mais ici le manque est compensé par l'influence de la ressource. Cet état peut aussi refléter le cas où l'individu remplit le minimum attendu en fiabilité, mais le minimum de ressource ne lui n'apporte aucun bénéfice.
Good : Cet état reflète le cas où l'individu remplit le minimum attendu en fiabilité et la ressource vient ajouter un bénéfice. Cela diminuera le total du niveau de vulnérabilité de 1.

6.2.3.4 Tables de probabilités

Les règles présentées ci-dessus permettent de décrire les conditions d'influence entre facteurs et les différents états que chaque nœud peut avoir. Afin d'utiliser ces règles pour déterminer la vulnérabilité, il est nécessaire de déterminer les probabilités associées à cette influence. Dans un premier temps, il convient donc de venir remplir la table de probabilité avec tous les états possibles pour chaque combinaison possible de chacun des nœuds. Le résultat de cette opération est que pour chaque combinaison de facteurs en amont d'un nœud nous aurons l'état prédominant correspondant à cette combinaison. Nous pouvons prendre comme exemple la figure 6.5 qui vient illustrer tous les états que chaque facteur jouant un rôle dans le nœud `CMES` peut générer. Dans cet exemple nous pouvons partir de l'hypothèse que l'acteur correspond à l'état émotionnel stable qui est demandé. La communication est active. Et le management est de type `CB`. Tous ces facteurs nous indiquent, via les règles d'influence, que l'état du nœud `CMES` sera à `Good`.

Dans un deuxième temps, il convient de déterminer les valeurs des tables de probabilités pour chaque nœud et pour chaque état de ce nœud. A ce stade, nous proposons deux approches. La première consiste à utiliser un ensemble de valeurs que l'on considérera comme étant des valeurs par défaut. En effet, ces valeurs ont été identifiées et position-

nées en fonction de la littérature. Elles sont donc génériques. Cependant, une entreprise ou une institution qui posséderait des valeurs plus spécifiques, construites par exemple sur des retours d'expériences, pourrait redéfinir les valeurs des tables de probabilités de manière à permettre des évaluations plus précises.

Dans le cas de l'approche générique, nous avons eu la démarche suivante : nous avons choisi de mettre à une valeur de 100 % la probabilité d'une influence à chaque fois qu'elle a respecté les règles que nous avons pu extraire de la littérature. Cela signifie que chaque fois que la règle se vérifie, on considère qu'elle a 100% de chance de se produire. Cette table de probabilité, qui semble caricaturale, a eu pour objectif, lors du développement de notre approche, de mettre en évidence la pertinence des influences que nous avions identifiées dans l'évaluation des vulnérabilités. Une fois cela validé, en accord avec notre partenaire industriel, nous avons décidé d'en faire le paramétrage par défaut du réseau bayésien. Ainsi, dans le pire des cas, le réseau bayésien identifiera des vulnérabilité humaines dans un SoSTS avec un taux de faux positifs relativement important. Nous jugeons cela préférable à une démarche inverse qui par défaut risquerait de ne pas identifier toutes les vulnérabilités possibles. Nous avons mené les cas d'études que nous présentons en chapitre 7 avec ces tables de probabilités.

La deuxième approche que nous proposons, consiste à permettre la modification des tables de probabilités ad-hoc, en fonction des données dont peuvent disposer les ingénieurs qui mettraient en œuvre notre approche. Il s'agit donc de permettre l'utilisation de tables de probabilités plus proches de la réalité, même si parfois incomplètes. En effet si nous avons pu extraire de la littérature les influences entre les facteurs, nous n'avons pas toujours pu renseigner avec précision la probabilité correspondant à ces règles. Ainsi, dans la mesure où les utilisateurs de notre approche pourraient avoir des données plus précises, il sont en mesure de modifier les tables proposées initialement avec leur propres valeurs. Typiquement, au lieu d'utiliser une probabilité de 100% lorsqu'une influence se révèle positive ou négative, un retour d'expérience pourrait indiquer que dans tel ou tel cas, la probabilité est plutôt de 86%. Prendre en compte cette valeur permettra au réseau bayésien de devenir de plus en plus précis au fur et à mesure que les probabilités par défaut sont remplacées par des valeurs bâties sur des données plus précises.

Enfin, pour ce qui concerne le calcul final du niveau de vulnérabilité, qui est réalisé sur le nœud final de notre réseau, nous avons défini sa table de probabilités (voir tableau 6.1). Celle-ci a pour but de retranscrire l'écart entre le profil idéal (du rôle) et le profil estimé d'un individu (de l'acteur). Plus cette différence est importante, plus la vulnérabilité

peut être potentiellement grave. Il convient d'ajouter à cela les influences des facteurs indirects. Tous ces éléments sont liés au nœud central de vulnérabilité qui retranscrit le niveau de vulnérabilité de l'individu. Ce niveau de vulnérabilité est noté de 0 à 5 avec une probabilité qui lui est associée. Les probabilités données dans la table 6.1 sont les probabilités par défaut et sont conditionnées aux probabilités des nœuds intermédiaires, cela afin de donner une probabilité et un niveau de vulnérabilité correspondant à l'état général de l'écart de l'individu (l'acteur) vis-à-vis du profil idéal (le rôle).

Niveau de vulnérabilité / *Point d'écart*	0	1	2	3	4	5
0	100%	0%	0%	0%	0%	0%
1	40%	60%	0%	0%	0%	0%
2	20%	60%	20%	0%	0%	0%
3	10%	80%	10%	0%	0%	0%
4	0%	20%	60%	20%	0%	0%
5	0%	10%	80%	10%	0%	0%
6	0%	0%	20%	60%	20%	0%
7	0%	0%	10%	80%	10%	0%
8	0%	0%	0%	20%	60%	20%
9+	0%	0%	0%	0%	40%	60%

TABLE 6.1 – Représentation simplifiée de la table de probabilité du nœud `Vulnerability Level`

Au final, l'approche que nous proposons pour estimer le niveau de vulnérabilité humaine d'un acteur dans un SoSTS est donc constitué d'un niveau de vulnérabilité associé à une probabilité que ce niveau soit atteint. Cependant, ce couple vulnérabilité - probabilité n'est pas suffisant pour permettre à un architecte d'estimer la potentielle dangerosité générée par cette vulnérabilité. Il faut ajouter au raisonnement l'estimation de l'impact de cette vulnérabilité. En effet, le fait qu'un individu soit fortement vulnérable, mais qu'il n'exerce aucune tâche critique, ne présente pas forcément un risque pour le système. Il faut donc ajouter à l'approche que nous proposons un moyen de déterminer l'impact d'une vulnérabilité sur un SoSTS. Par ailleurs, un deuxième point est important pour permettre une vue complète de l'impact d'une vulnérabilité identifiée sur un système. Il s'agit de la manière dont celle-ci risque de s'étendre au reste du système.

6.3 Propagation et impact de la vulnérabilité

La propagation d'une vulnérabilité humaine au sein d'un SoSTS est l'action qui consiste à transmettre cette vulnérabilité d'un individu source à un autre. C'est par exemple le cas pour une vulnérabilité bien connue sous le terme de panique. Celle-ci va générer une contamination émotionnelle vers d'autres individus qui à leur tour vont eux même présenter cette vulnérabilité et peuvent alors la transmettre également. La transmission d'une fausse information à partir d'un individu vers un groupe avec lequel il est en contact est un deuxième exemple de propagation d'une vulnérabilité au sein d'un SoSTS. Cette propagation peut même être par moment plus nuisible pour le système que la vulnérabilité elle-même. En effet, si parfois une forte vulnérabilité portée par un individu peut potentiellement nuire au SoSTS, cela n'est pas forcément toujours le cas lorsque la vulnérabilité ne se propage pas. Ainsi, un individu isolé pouvant être fortement vulnérable, peut n'avoir qu'un faible impact sur les missions du SoSTS. A *contrario*, un individu avec un niveau de vulnérabilité moyen, mais une forte capacité à la transmettre peut avoir un impact extrêmement important sur le SoSTS. Une vulnérabilité ne se regarde donc pas seulement par son niveau et sa probabilité d'occurrence elle s'évalue aussi par sa capacité à se transmettre dans le système.

Dans cette section, nous abordons les modèles de propagation que nous avons mis en place pour simuler la propagation d'une vulnérabilité humaine dans un SoSTS. Nous détaillons en effet dans la suite deux exemples d'application de modèles de propagation. Le premier est basé sur une approche *ad-hoc* développée avec notre partenaire industriel. Le second utilise deux modèles différents issus des sciences humaines pour lesquels les mécanismes de propagation sont adaptés à la propagation d'une vulnérabilité dans un SoSTS.

6.3.1 Approche ad-hoc pour l'estimation de la propagation

Dans un premier temps, afin d'évaluer la propagation d'une vulnérabilité dans un SoSTS, nous avons construit une approche *ad-hoc*. Celle-ci a été réalisée avec l'aide de notre partenaire industriel. Nous avons pu ainsi, au fur et à mesure des échanges, créer une table de propagation basée sur le niveau de vulnérabilité d'un individu et avons également qualifié l'impact possible qu'il pourrait avoir sur ce SoSTS.

L'impact d'une vulnérabilité identifiée est un critère important pour en qualifier le risque. En effet, il ne suffit pas d'être vulnérable pour mettre en danger le SoSTS. Il faut

aussi que l'opérateur soit capable d'impacter son système, soit par ses liens fonctionnels, soit par sa position hiérarchique. Pour représenter cette impact nous avons choisi de calculer l'impact en fonction de la position hiérarchique définie dans les facteurs indirects du rôle auquel est lié l'acteur et de la confiance que détient cet acteur. La définition de cette règle est en Règle 6.6.

$$\boxed{Impact(A) = (PositionOf(A) + ConfidenceIn(A))/2} \tag{6.6}$$

Pour calculer l'impact de notre opérateur sur notre SoSTS nous considérons de manière équivalente la position hiérarchique et la confiance investie dans ce rôle par l'architecte. En effet, pour qu'un opérateur ait un impact important sur le SoSTS, il est nécessaire qu'il ait une position hiérarchique importante, mais aussi que le système ait un besoin de confiance envers cet opérateur. Cette confiance s'exprime par la manipulation d'informations sensibles ou de tâches essentielles au système.

Cette première approche a identifié trois types de liens permettant la propagation d'une vulnérabilité humaine dans un SoSTS. Les deux premiers sont les liens qui vont représenter la transmission d'informations et la délégation d'objectif. Les deux types de liens sont décrits dans le langage de manière structurelle (voir chapitre 5), car ils permettent de définir l'architecture du SoSTS. Le troisième lien est le lien relationnel entre individus. Il va permettre notamment la contamination émotionnelle. En effet, la contamination émotionnelle via le stress peut se dérouler lorsqu'il existe un lien personnel entre deux personnes [85].

Dans cette première approche, nous avons choisi de permettre la propagation de vulnérabilité dans tous les types de liens : les liens structurels faisant partie de la représentation du système, les liens individuels pouvant appartenir à chacun des opérateurs présents dans le système.

Tous ces éléments nous amènent à construire le tableau 6.2. Dans ce tableau on peut clairement distinguer les deux liens structurels. Ils ne peuvent générer de la contamination qu'en cas de lien direct. A l'inverse, le lien relationnel peut aussi bien impacter un individu de manière directe ou de manière indirecte. Cela représente la notion de relation personnelle entre individus. Notons que la contamination est possible même si dans l'architecture deux individus par lesquels elle se transmet ne sont pas liés.

Après l'expérimentation, réalisée avec notre partenaire industriel, les retours de leurs experts montrent que cette approche est générique et permet une évaluation approximative de la propagation. Cependant, cette approche ne donne pas la probabilité qualifiant le

A → B	***Direct Link***	***Indirect Link***
Objective delegation	impact(A) > 3 & vulnerability(A) > 3	N/A
Document transmission	impact(A) > 2 & vulnerability(A) > 2	N/A
Personal relationship	impact(A) > 2 & vulnerability(A) > 2 & CMES(B) ≠ *Good*	impact(A) > 2 & vulnerability(A) > 2 & CMES(B) ≠ *Good*

TABLE 6.2 – Conditions de contamination en fonction des liens entre opérateurs et rôles

risque de propagation et ne permet pas d'ajuster le modèle de propagation en fonction des types de menace. En effet, les règles ici présentées sont de type binaire. La probabilité est donc de un ou de zéro sans aucune progressivité.

6.3.2 Estimation de la propagation basée sur des modèles issus des sciences sociales

D'après les conclusions tirées de la première approche de propagation, nous avons cherché à améliorer notre modèle en permettant une adaptabilité de la propagation en fonction du type de scénario d'attaque que nous prenons en compte (type d'attaque particulier qui va cibler un individu vulnérable). En effet, la propagation de cyberattaques dans un SoSTS peut être différente selon le type d'attaque. Ainsi, de manière à tester la capacité de notre approche à adapter ses outils d'analyse en fonction des besoins, nous avons focalisé notre modèle de propagation sur deux grands types d'attaques :

Attaques via de fausses informations : Avec ce modèle, nous évaluons la capacité de notre approche à représenter la transmission de fausses informations venant d'un individu vulnérable vers le reste du SoSTS.

Attaques sur la contamination émotionnelle : Sur ce modèle nous cherchons à représenter la propagation d'un fort impact de stress sur un individu dans un SoSTS. Ce type d'événement peut arriver suite à une attaque qui génère une situation de gestion de crise par exemple.

6.3.2.1 Estimation de la propagation basée sur un modèle d'attaques de fausses informations

Nous avons sélectionné au sein de la littérature le modèle présenté dans [86] qui permet de prendre en compte la propagation d'une fausse information dans un SoSTS. Ce modèle

décrit ici la propagation de contenus viraux dans différents réseaux sociaux et permet de donner une probabilité de transmission de ce type de contenus à travers différentes organisations humaines. Ainsi, ce modèle nous permet de représenter de manière générique la propagation d'une fausse information dans un groupe d'individus.

A partir de [86], nous avons adapté le modèle proposé à notre approche. Sa prise en compte dans notre estimation peut se traduire à la manière du tableau 6.3. Ce tableau permet la représentation de la propagation d'une fausse information selon l'impact d'un individu dans un système. Pour permettre une meilleure prise en compte de ce modèle de propagation, nous l'avons lié au type de lien correspondant à la transmission de documents dans notre langage. En effet, on considère que c'est à partir de ce lien qu'une attaque peut se propager. Si un opérateur est contaminé par une fausse information, c'est par ce lien qu'il va pouvoir la transmettre à d'autres opérateurs du système.

Impact & vulnerability	***Document transmission first link (B)***	***Document transmission second (C)***
impact(A) > 2 & vulnerability(A) > 2	18%	4%
impact(A) > 3 & vulnerability(A) > 3	75%	25%
impact(A) > 4 & vulnerability(A) > 4	99 %	50%

TABLE 6.3 – Contamination entre opérateurs via le lien de transmission de documents.

Pour obtenir les probabilités présentes dans le tableau 6.3, nous sommes partis des données venant de [86] et synthétisés dans la conférence sur ce papier [1]. Nous avons considéré le cas dans lequel un attaquant aurait réussi à contaminer au moins une personne (événement A). Nous avons conditionné notre probabilité de contaminer directement d'autres individus à ce premier postulat (événement B). Ce qui nous donne pour la première colonne de la première ligne (6.7)

$$\boxed{P(A|B) = \frac{1.2}{6.8} = 17.8\%} \tag{6.7}$$

Pour la deuxième colonne, nous partons du même évènement qui sert de contexte (l'évènement A). Pour l'événement C nous prenons le cas où il y a une contamination d'un individu provoqué par un individu déjà contaminé. Ce qui nous donne pour la deuxième colonne de la première ligne (6.8)

$$\boxed{P(A|C) = \frac{0,3}{6,8} = 4.4\%} \tag{6.8}$$

1. https ://www.youtube.com/watch ?v=gykNdC2CLVg

Pour la deuxième ligne, nous avons modifié le contexte en le concentrant uniquement sur des cas où l'attaquant peut contaminer d'autres personnes que la première personne contaminée. Pour la troisième ligne, nous avons considéré que la contamination du premier cercle était quasiment certaine et avons choisi comme contexte le cas où les personnes déjà contaminées peuvent contaminer d'autres personnes.

Pour établir les règles d'impact et de vulnérabilité présentes dans le tableau 6.3, nous sommes partis des règles issues du modèle *ad-hoc*, que nous avions validés au préalable au moyen de différents cas d'études qui seront présentés au chapitre suivant. Ces règles permettent une détection de la vulnérabilité binaire : soit il y a vulnérabilité, soit pas. L'apport de ce nouveau modèle de propagation est justement sur la capacité qu'il amène à fournir des probabilités progressives. Ainsi, si l'on regarde la première ligne du tableau, il n'y a que 18% de chance qu'une propagation soit possible entre un opérateur touché par une désinformation et l'opérateur avec lequel il a un premier niveau de relation sociale. Cette probabilité tombe à 4 % pour l'opérateur de second niveau de relation sociale en cas de rebond d'une contamination de l'opérateur de premier niveau de relation sociale vers ce même opérateur.

De manière plus claire, pour une attaque de désinformation sur un opérateur A, pour lequel l'évaluation de vulnérabilité identifie un impact de niveau 2 et une vulnérabilité de niveau 2, et dans le cas où cet opérateur est en relation avec un opérateur B à travers une transmission de documents, alors les chances que la vulnérabilité soit propagée à l'opérateur B sont de 18 %. Si l'opérateur B était lui identifié comme vulnérable dans les mêmes conditions, alors l'opérateur C avec qui il aurait lui-même une relation de transmission de documents aurait une probabilité de 4% d'être contaminé à son tour. On peut donc considérer que même si l'opérateur A est identifié comme vulnérable, il peut ne pas être considéré comme un vecteur important de propagation. *A contrario*, la deuxième ligne du tableau vient illustrer que l'opérateur A, s'il est identifié vulnérable, alors il a une majorité de chances de propager la vulnérabilité à un autre individu. Enfin la troisième ligne vient donner une quasi-certitude que l'opérateur A va propager la vulnérabilité vers l'opérateur B, qui a sont tour à 50 % de chances de contaminer un troisième individu.

6.3.2.2 Estimation de la propagation basée sur un modèle d'attaques de type contamination émotionnelle

Le deuxième modèle de propagation que nous avons développé est celui lié à la contamination émotionnelle. En effet, capitaliser sur la panique d'un opérateur peut être un

bon moyen pour un attaquant de nuire à un SoSTS par exemple. Le modèle de propagation que nous avons choisi pour représenter cette contamination émotionnelle dans un système est extrait de [87]. Ce modèle nous a permis de créer la table 6.4 qui vient donner cette probabilité de contamination soit directement dans l'architecture du SoSTS par l'utilisation d'un lien de type délégation d'objectifs entre deux opérateurs, soit par un lien personnel entre deux individus mais hors architecture. En effet, Alice et Bob sont en relation à travers l'architecture du SoSTS. Mais ils peuvent également avoir une relation hors de ce système. Le choix de ces deux liens pour ce type de contamination est dû au fait que, dans les deux cas, les risques d'une contamination émotionnelle sont importants

Impact & vulnerability	***Delegation or personal link first link***	***Delegation or personal link second***
impact(A) > 2 & vulnerability(A) > 2	62.71%	46.50%
impact(A) > 3 & vulnerability(A) > 3	76.49%	62.71%
impact(A) > 4 & vulnerability(A) > 4	86.29%	76.49%

TABLE 6.4 – Probabilité de contamination entre opérateurs de premier et de second niveau

Les probabilités que nous avons identifiées pour estimer cette propagation sont présentées dans le tableau 6.4. Ces probabilités correspondent ici à une situation dans laquelle les conditions de stress sont élevées sur les individus. Pour transposer le modèle de propagation proposé par [87] au modèle de facteurs humains que nous proposons, nous avons appliqué ce que les auteurs identifient comme vecteur de propagation, à savoir le concept de pression sociale qui est une métrique s'échelonnant de 1 à 5 , à celui de vulnérabilité humaine. Ainsi, nous considérons que la pression sociale, concept sur lequel s'appuient les auteurs de [87] pour estimer la probabilité d'une contamination, est similaire à ce que véhiculerait un opérateur touché par une vulnérabilité dans une situation de stress important. Par exemple, dans le cas où un opérateur identifie qu'il est victime d'une cyber-attaque. Pour déterminer les valeurs des probabilités, nous avons utilisé l'équation 6.9 suivante venant de [87].

$$\boxed{\frac{1}{(1+e^{-0.66*v+0.80})}} \tag{6.9}$$

Ce nouveau modèle de propagation est proche des résultats obtenus avec le modèle *ad-hoc*. Il va cependant permettre d'adapter l'estimation de la propagation en fonction du type des menaces. Il est donc possible d'identifier et de proposer d'autres modèles de propagations à utiliser lors de l'estimation de la propagation d'une vulnérabilité en

fonction d'un type d'attaque spécifique. A terme, on peut considérer que les utilisateurs de notre approche pourraient développer de tels modèles d'estimation en fonction de leur connaissance dans leur domaine et/ou de leurs retours d'expériences, de manière à enrichir les capacités de détection de propagation.

Pour illustrer les deux types de propagation, nous proposons un exemple visible sur la figure 6.6. Cet architecture de SoSTS a été développée avec HoS-ML Editor. Il s'agit de l'architecture d'un des cas d'études que nous avons réalisés (voir chapitre 7) pour valider notre approche.

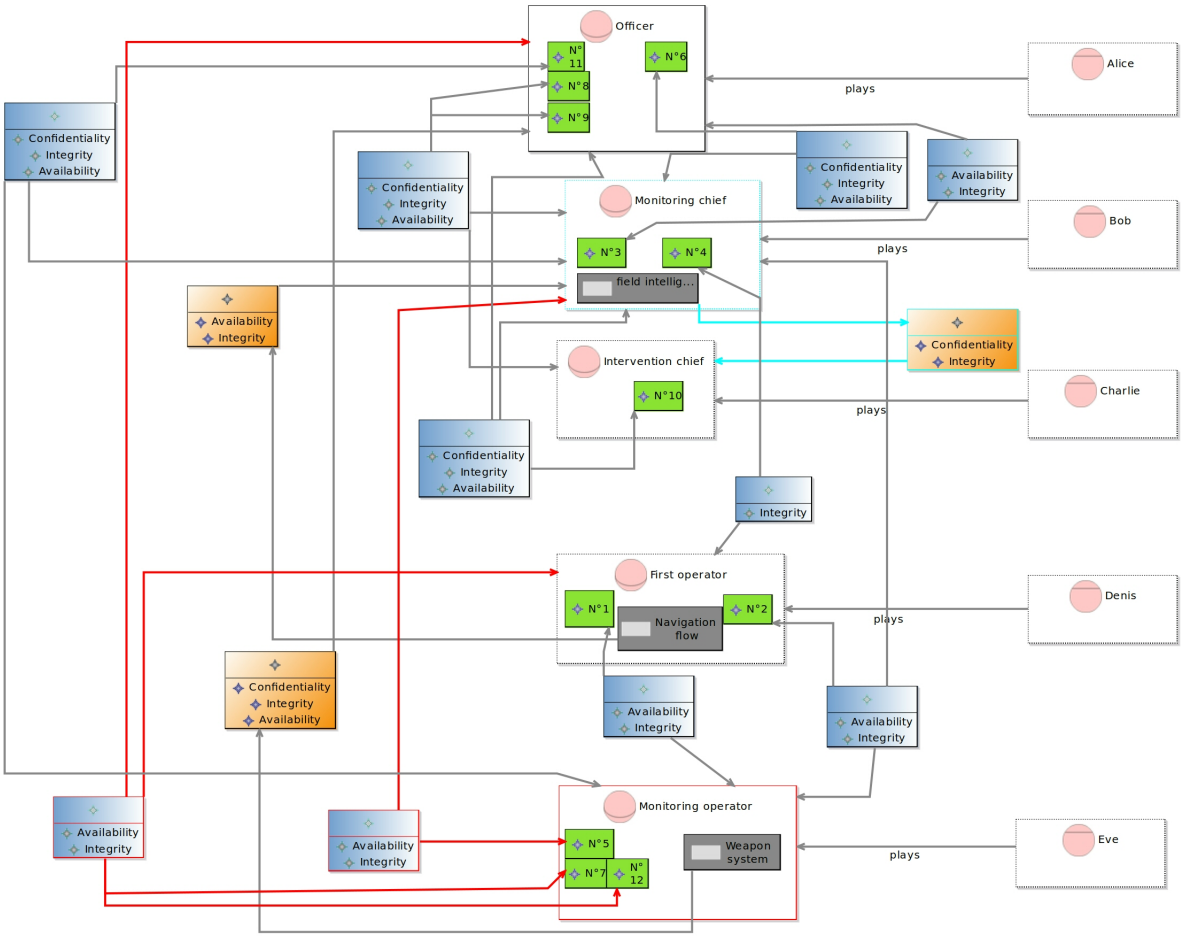


FIGURE 6.6 – Exemple d'architecture avec deux propagations de vulnérabilité possibles

Nous avons généré ici des scénarios différents, chacun ayant pour but de venir illustrer un type de propagation spécifique. Dans le premier scénario, nous avons simulé une attaque liée à de fausses informations qui pourraient être transmises par un opérateur à un autre.

Dans ce cas précis, l'opérateur sera face à de fausses données qui lui sont remontées par les systèmes physiques qu'il utilise (ici de fausse trame AIS par exemple). Le second scénario représente un cas de forte transmission émotionnelle, ce qui dans notre cas correspond à une attaque liée à un chantage sur un individu générant un stress qui à long terme peut contaminer d'autres personnes. Ces propagation sont visibles sur la figure 6.6 : en rouge la première propagation liée au chantage et qui va générer une possible contamination sociale ; en cyan le second scénario sur fausses informations. Ici nous avons représenté la propagation la plus probable retournée par les données. Les données qui ont permis de créer ces scénarios sont visibles dans le chapitre 7. Ce sont les mêmes scénarios utilisés dans ce cas de propagation.

6.4 Implémentation de l'estimation de la vulnérabilité dans HoS-ML Editor

Pour permettre à un architecte d'utiliser l'estimation de la vulnérabilité dans un SoSTS et de mesurer l'impact de cette vulnérabilité dans cette architecture. Nous avons implémenté le réseau bayésien dans notre logiciel en lui associant une fonction permettant de calculer automatiquement la vulnérabilité d'un opérateur en fonction du rôle qu'il occupe s'il est ciblé par une menace. Pour compléter cette estimation de la vulnérabilité, nous avons implémenté dans le logiciel le retour visuel de l'impact sur le SoSTS de la vulnérabilité avec un retour utilisateur textuel permettant de comprendre la vulnérabilité.

Pour implémenter le réseau bayésien, nous avons fait le choix d'utiliser les librairies *amidst* [2] qui permettent directement d'implémenter notre réseau bayésien et ses données sous forme de différentes tables de probabilité de manière simple, cela dans le langage utilisé par Sirius : Java. Cette librairie est d'autant plus intéressante qu'elle permet de paralléliser les calculs lors de la résolution du réseau bayésien. Cela permet de gagner un temps de calcul raisonnable comparé aux autres librairies que nous avons pu tester.

En complément du réseau bayésien, nous avons ajouté dans le logiciel une matrice de pondération. Cette matrice de pondération permet directement à l'architecte de venir modifier le poids de chaque facteur dans le calcul de la vulnérabilité par le réseau bayésien. Cette modification de poids n'est qu'une simple multiplication du niveau de vulnérabilité que chaque facteur peut apporter sur le nœud final du réseau bayésien. L'objectif de cette

2. http ://www.amidsttoolbox.com/

matrice est de permettre à l'architecte de venir personnaliser le réseau bayésien avec sa connaissance métier , même s'il ne possède pas de données métier réelles. Pour finir, nous avons implémenté une fonction de recherche automatique de vulnérabilité. Cette fonction a pour but d'aider l'architecte à trouver plus facilement une vulnérabilité potentielle dans son architecture. Pour ce faire un architecte doit renseigner deux éléments qui sont :

— Le nombre de facteurs différents que l'architecte accepte dans la recherche.
— Le décalage maximum que peut avoir chacun des facteurs vis-à-vis du facteur idéal.

Ces deux éléments vont permettre à l'algorithme de tester toutes les combinaisons, avec le nombre de facteurs différents définis par l'architecte et le décalage inférieur ou égal à ce qu'a définit l'architecte pour chaque facteur. L' algorithme informera au final le cas où la vulnérabilité la plus importante a été trouvée avec ces paramètres. Le temps de calcul de cette fonction reste peu coûteux, puisque le nombre maximal de combinaisons possibles sur tous les facteurs humains et de 250 000. Ce nombre de combinaisons reste calculable pour le logiciel avec un ordre de grandeur de plusieurs dizaines de minutes. Cette fonction va permettre de trouver facilement la vulnérabilité la plus importante dans un contexte donné.

Pour l'implémentation de la propagation, nous avons dans un premier temps mis en place le modèle *ad-hoc* à travers la coloration des différents éléments touchés par la vulnérabilité.

Dans un second temps, nous avons mis en place les modèles de propagation détaillés dans la Section 6.3. pour mettre en place ces modèles de propagation, il a fallu ajouter un paramètre de personnalisation à l'élément menace pour que l'architecte puisse choisir entre les deux types de propagation. L'élément menace a ainsi trois configurations possibles, la configuration *ad-hoc* permettant de visualiser l'impact *ad-hoc*. La configuration "ingénierie sociale" permet de visualiser l'impact à travers la contagion émotionnelle et la configuration "désinformation" permet de visualiser l'impact d'une fausse information sur le système.

Enfin, suite à la simulation, de manière à comprendre la propagation, le logiciel fournit à l'architecte un élément récapitulatif basé sur les informations suivantes :

— L'opérateur et le rôle source de la propagation.
— Le niveau de vulnérabilité et la probabilité de celle-ci.
— Le nombre de facteurs directs de l'opérateur, différent vis-à-vis du profil idéal attendu par le rôle.

6.5 Résumé du chapitre

Dans ce chapitre, nous avons proposé la définition d'un moyen d'évaluation de la vulnérabilité humaine dans un SoSTS. Cette estimation permet à un architecte de venir évaluer le niveau de vulnérabilité humaine dans l'architecture ainsi que son potentiel impact. Cette définition repose sur plusieurs contributions que nous avons détaillées.

La première contribution que nous avons développée est un processus d'estimation de la vulnérabilité humaine dans les SoSTS. Pour évaluer la vulnérabilité humaine dans un SoSTS le processus mis en place permet l'usage du langage HoS-ML ainsi que de l'outil HoS-ML Editor. Le processus se déroule en quatre étapes en partant de la modélisation des rôles du SoSTS vers la simulation de la vulnérabilité dans le SoSTS.

La deuxième contribution est la définition d'une méthode d'estimation de la vulnérabilité humaine. Cette méthode consiste en la comparaison du rôle idéal qui va être occupé par un opérateur humain entre les facteurs directs attendus sur le rôle et les facteurs qu'aura réellement l'opérateur humain. Cette différence entre l'idéal et l'attendu, étant aussi influencée par les facteurs indirects. Pour permettre de calculer une estimation de cette différence avec l'influence des facteurs indirects, nous avons utilisé une approche probabiliste à travers un réseau bayésien. Ce réseau bayésien permet à partir des éléments de la littérature ainsi que de règles que nous ayons traduit de la littérature, de proposer des conjectures et de donner une estimation probabiliste de la vulnérabilité d'un opérateur en fonction de la différence entre facteurs directs idéaux et attendus, ainsi que de l'influence des facteurs indirects.

La troisième contribution est une proposition de modèle de propagation de la vulnérabilité humaine dans un SoSTS. Le modèle de propagation que nous proposons utilise les liens entre opérateurs humains. En fonction du type de lien, la propagation de la vulnérabilité ne sera pas la même. Il y a deux types de propagation, ce qui correspond au nombre de liens structurels dans l'architecture. Le premier type de propagation est lié au document et le deuxième aux objectifs. Pour chacune de ces propagations, nous avons été chercher dans la littérature des modèles de propagation correspondant et permettant d'estimer la propagation d'un opérateur en fonction de ce qui le relie à un opérateur dans l'architecture.

Dans le chapitre suivant, nous allons voir comment utiliser les contributions de ce chapitre et de celui d'avant dans des cas d'études industrielles.

Chapitre 7

Validation de l'approche

Sommaire

Les chapitres antérieurs ont permis d'exposer les différentes concepts, modèles et outils que nous avons élaborés au cours de nos travaux de recherche. Plus particulièrement,

nous avons pu décrire la manière dont nous estimions la vulnérabilité humaine dans une architecture de SoSTS. Pour cela, nous avons défini un métamodèle permettant de caractériser les propriétés d'un humain à prendre en compte vis-à-vis de notre problématique. Puis, nous avons développé le langage HoS-ML permettant de décrire une architecture humaine dans un SoSTS. Nous avons également spécifié un moyen d'évaluer la vulnérabilité de l'architecture décrite à l'aide de HoS-ML en utilisant un réseau bayésien. Puis, nous nous sommes intéressés au moyen d'évaluer le risque de propagation d'une vulnérabilité identifiée dans le reste de l'architecture décrite. Nous avons intégré tous ces éléments au sein d'un outil en utilisant une approche basée sur les modèles. Enfin, nous avons décrit une méthodologie guidant l'usage de l'outil et du langage HoS-ML afin de décrire l'architecture d'un SoSTS et de tester différents scénarios de vulnérabilité sur l'architecture donnée.

Suite à cela, nous avons cherché à valider notre approche. Pour ce faire, nous avons utilisé une validation par cas d'étude [88] afin de confronter l'utilisation de notre approche à des modélisations de cas d'étude réels. Les résultats ainsi obtenus ont été ensuite soumis à des experts en architecture et à des experts en cyberdéfense.

Ce chapitre présente la méthodologie que nous avons suivie pour la validation de notre approche, ainsi que les cas d'étude réels qui nous ont été fournis par notre partenaire industriel. Nous présentons ensuite les retours que nous ont faits les différents experts. Pour finir, nous discutons des conclusions à tirer des résultats de cette étude.

7.1 Méthodologie des cas d'études

Pour valider notre approche, nous avons choisi une méthodologie par cas d'études [88]. Cette solution permet de mettre en place une méthodologie rigoureuse pour encadrer les expérimentations nécessaires à la validation, ainsi que les biais potentiels qui pourraient se produire dans celles-ci. Cette méthode se base elle-même sur des méthodes antérieures issues des sciences humaines, telle que la méthodologie de Yin [89] faisant référence dans l'évaluation de modèles impliquant l'humain. La méthode de validation par cas d'étude repose sur les étapes suivantes :

— Définition du contexte de l'étude
— Conception des cas d'étude
— Sélection des cas d'étude
— Définition des procédures d'évaluation et des différents rôles dans les cas d'étude

— Collecte des données issues des cas d'étude
— Analyse des résultats
— Validation des résultats

7.1.1 Contexte

Dans un premier temps, nous avons défini les questions de recherche auxquelles devaient répondre ces cas d'études. Elles ont été mises en place avec notre partenaire industriel, afin de permettre à celui-ci d'évaluer l'intérêt de notre approche. Les questions suivantes ont été retenues :

QR1C : Comment utiliser HoS-ML dans la définitions d'architectures pouvant être complexes et portant sur des domaines différents ?

QR2C : Les résultats simulés sont-ils pertinents et permettent-ils de détecter des vulnérabilités ?

Les études de cas sur lesquelles nous avons appliqué ces questions de recherche portent sur des domaines différents. La première étude de cas concerne la piraterie maritime. Elle décrit un navire luttant contre des actions de piraterie maritime dans un détroit. La seconde étude de cas concerne le contrôle aérien dans un aéroport confronté à une menace terroriste.

7.1.2 Conception des cas d'étude

L'étape suivante, portée par la méthodologie de validation, consiste à concevoir les études de cas supports à l'évaluation. Pour cela, il convient d'identifier dans un premier temps le type de "*case and analysis unit*" correspondant aux cas d'études visés. Les différentes catégories sont définies dans [89]. Nous avons deux types de concepts qui sont le *single-case designs* et le *multiple-case designs*. Ces deux concepts définissent si l'étude comprend un ou plusieurs cas d'études. Nous avons aussi deux types d'unité d'analyse qui sont *holistic* et *embedded* qui vont définir si pour chaque cas d'étude il y a une analyse d'un ou de plusieurs éléments la composant. En accord avec notre partenaire industriel, nous avons choisi de mener une validation reposant sur plusieurs cas d'études, permettant d'évaluer ainsi plusieurs architectures. Ces différentes architectures permettent ainsi d'évaluer la capacité de notre approche à s'adapter à plusieurs situations. Au sens de la méthodologie, il s'agit d'une étude de type *multiple-case*. Pour ce qui est de "*l'analyse*

unit" qui définit le nombre d'éléments que nous analysons pour chaque cas d'étude, notre évaluation entre dans un cadre *embedded design*. En effet, nous analysons plusieurs scénarios par cas d'étude (en opposition à un cadre holistique dans lequel il n'y a qu'un scénario par cas d'étude).

Afin de bien concevoir les études de cas, il convient de bien définir les objectifs que l'on cherche à atteindre en les réalisant. Nous avons identifié les objectifs suivants :

— Être capable de définir les différents profils humains venant caractériser les rôles du cas d'étude à définir. Cela permettra d'avoir une meilleure vision des rôles à modéliser d'une part, et d'autre part des exigences de sécurité liées à la transmission des informations liées à ces rôles ;

— Être capable de faire évaluer par des experts les résultats de l'expérimentation. C'est-à-dire de prendre en compte la (ou les) potentielle(s) vulnérabilité(s) humaine(s) identifiée(s) dans les différents cas, de juger de leur pertinence, et d'évaluer si ces résultats peuvent à terme mener à une correction de la (ou des) potentielle(s) vulnérabilité(s) identifiée(s).

7.1.3 Sélection des cas d'étude

Dans cette étape, il s'agit d'identifier et/ou de définir les cas d'étude à évaluer au cours de l'étude. Nous nous sommes appuyés sur les critères suivants :

— Les cas doivent être réalistes et permettre une représentation d'un SoSTS ;

— La complexité des cas d'étude ne doit pas être trop importante pour permettre une représentation visuelle aisément interprétable ;

— Les cas d'étude étant des cas industriels, ils doivent permettre une publication même en étant privés d'informations sensibles ;

— L'équipe industrielle et opérationnelle doit nous donner les éléments métier et les éléments d'architecture permettant de définir les différents cas. Cela passe notamment par la transmission des différentes tâches liées à chacun des rôles, ainsi que par une définition des rôles et de leurs profils humains associés.

Deux cas d'étude ont ainsi été retenus. Le premier est purement du domaine naval et concerne la lutte contre la piraterie maritime. Le second s'intéresse à un autre domaine, le contrôle aérien, ce qui permet de montrer l'applicabilité de notre approche hors du contexte naval. Les deux cas sont présentés et discutés plus loin dans ce chapitre. Dans les deux cas, nous gérons la complexité en nous limitant à la modélisation d'une chaîne fonctionnelle par cas d'étude et non pas en modélisant la totalité des chaînes fonctionnelles

du SoSTS considéré.

7.1.4 Les procédures et rôles

Cette étape a pour objectif d'identifier les responsabilités des différents membres de l'équipe d'évaluation dans la conduite de cette dernière. Nous avons ainsi identifié 3 équipes distinctes.

L'équipe « industrielle », dirigée ici par David Hairion, fournit les éléments d'architecture des cas d'étude. Elle doit aussi fournir différents éléments nécessaires aux cas d'étude tels que les objectifs et buts de chaque rôle, la répartition des tâches entre les rôles ainsi que leurs liens informationnels. Deux autres experts de cette équipe donneront aussi leur avis sur les résultats obtenus.

L'équipe « opérationnelle », composée d'officiers de la marine spécialisés dans la cyberdéfense, doit fournir pour chaque étude de cas les profils humains de chacun des rôles. Elle doit également donner son avis en tant qu'experts opérationnels sur les vulnérabilités identifiées par notre approche.

L'équipe nommée « HOS-ML », composée de Nicolas Belloir et de moi-même, a pour objectif de réaliser les cas d'étude avec les données fournies par les deux équipes précédentes, puis de soumettre les modèles réalisés et les résultats des simulations à leur validation.

7.1.5 Collecte des données

La collecte des données concerne deux phases de la campagne d'évaluation. En effet, certaines données sont nécessaires à la construction des cas d'étude. D'autres sont les résultats des cas d'étude et sont nécessaires à la validation de ces derniers. Pour la construction des cas d'étude, nous avons donc dû collecter des données auprès de chacune des équipes. Ces données ne contiennent aucune donnée personnelle puisqu'elles concernent des profils types. De plus nous avons expurgé ces données de tout élément sensible.

Les données qui ont donc été collectées auprès de l'équipe industrielle sont les données suivantes :

— Les différents objectifs de chaque rôle ;
— Les différentes informations que détient chaque rôle ;
— Les liens entre rôles vis-à-vis des objectifs liés aux informations qui circulent ;
— Le contexte dans lequel les cas d'étude sont opérés.

Les données qui ont été collectées auprès de l'équipe opérationnelle sont les données suivantes :

— les facteurs directs caractérisant les rôles à modéliser ;

— Les facteurs indirects caractérisant ces mêmes rôles ;

— Les procédures à prendre en compte dans ces cas d'étude face à une menace cyber.

Les données résultant de la simulation ont été collectées de manière à être traitées à l'étape suivante.

7.1.6 Analyse des résultats

L'étape d'analyse consiste à évaluer les cas d'étude et leurs résultats de manière à en tirer des conclusions. Pour cela, nous avons, à la fois, traité les modèles réalisés et les résultats fournis par l'outil après simulation. Nous avons livré ces données aux experts pour qu'ils puissent évaluer le réalisme des modèles ainsi que les résultats fournis par notre approche. Ces experts sont les membres de l'équipe industrielle et de l'équipe opérationnelle.

L'avis des experts permettra de répondre aux questions de recherche identifiées en Section 7.1.1. Plus particulièrement :

— Les retours des experts en architecture permettront de répondre à la question QR1C portant sur les capacités de modélisation de HoS-ML ;

— le retour des experts vis-à-vis des résultats amenés par l'outil nous fournira des éléments pour répondre à la question QR2C.

7.1.7 Validité des résultats

Afin d'assurer la validité des résultats de notre étude, nous avons choisi de nous reposer sur les experts des équipes industrielle et opérationnelle. Ils sont au nombre de 4 et sont répartis de manière égale dans les deux équipes. Les experts de l'équipe industrielle sont des architectes seniors, spécialisés dans les systèmes de systèmes navals. Les experts opérationnels sont également des experts seniors, spécialisés dans la cyberdéfense opérationnelle des systèmes navals.

Afin de collecter leurs avis, nous avons créé deux questionnaires en vue de les interviewer. Le premier est à destination des architectes et doit permettre de répondre à notre QR1C. Le second est à destination des deux équipes. Il a pour but de répondre à QR2C.

Pour construire les questionnaires dédiés aux experts, nous nous sommes basés sur

Numéro	Question
1	Les éléments de contexte du scénario vous semblent-ils réalistes ?
2	L'attaque sur le sujet vous semble-t'elle probable ?
3	Sur une échelle de un à cinq, évaluer sa probabilité (un étant non probable et cinq probable)
4	l'attaque sur le sujet vous semble-t-elle réaliste ?
5	La vulnérabilité identifiée par la méthode vous semble-t-elle réaliste ?
6	Le niveau qui lui est associé vous semble-t-il correct ?
7	La probabilité qui lui est associée vous semble-t-elle correcte ?

TABLE 7.1 – Questionnaire destiné à tous les experts

l'approche du MITRE [90] et du SANS [91]. Ces deux instituts sont spécialisés sur le domaine de la cybersécurité et ont l'habitude de produire des rapports en questionnant des experts. L'approche qui est proposée par ces instituts permet de faire ressortir les métriques importantes en termes de cybersécurité.

La méthode employée a été la suivante. Nous avons, à partir des QRC identifiées, cherché dans ces deux sources les métriques utiles à l'évaluation de notre approche. Une fois ces métriques identifiées, nous avons créé les questionnaires. Les deux questionnaires sont visibles dans les tableaux 7.1 et 7.2.

Afin de permettre l'audition des experts en limitant les biais, nous avons suivi les recommandations présentes dans [89]. Nous avons notamment toujours auditionné les experts individuellement. Nous avons veillé à ce que les experts n'échangent pas autour des cas d'étude avant leur audition.

Enfin nous avons toujours utilisé la même trame pour chaque expert :

— Présentation de la thèse
— Présentation d'un scénario et de ses données
— Pour chaque scénario, soumettre l'expert au questionnaire liée a la cybersécurité
— Dans le cas d'un expert architecte lui soumettre le questionnaire sur l'architecture

7.1.8 Limitation de la validation

Malgré l'usage d'une méthodologie prévue pour des validations, notre étude présente plusieurs limitations :

Confidentialité des cas d'étude : Dans les scénarios des différents cas d'étude, les éléments ont été blanchis pour permettre de préserver la confidentialité industrielle de notre partenaire ainsi que certains éléments sensibles pouvant apparaître en lien

Numéro	Question
1	Le langage HoS-ML permet-il de représenter les différents acteurs présents dans le cas d'étude ?
2	Le langage HoS-ML permet-il de représenter les différents rôles identifiés dans le cas d'études ?
3	Le langage HoS-ML permet-il de représenter les différents objectifs de chaque rôles ?
4	Le langage HoS-ML permet-il de représenter les différentes données utilisées par les rôles ?
5	Le langage HoS-ML permet-il de représenter suffisamment d'éléments pour définir un SoSTS ?

TABLE 7.2 – Questionnaire destiné aux experts architectes

avec certains systèmes. Nous avons essayé de préserver au maximum la transposition des vulnérabilités dans les cas blanchis ainsi que la cohérence de l'avis des experts sur ces cas.

Échelle d'expertise : Nous avons fait le choix de prendre peu d'experts pour évaluer les cas d'étude, mais de prendre des experts seniors. Il faudrait, à terme, un plus grand panel d'experts pour permettre une évaluation plus fine de l'approche.

Fiabilité des données : Les données recueillies pour construire les cas d'étude l'ont été auprès d'une seule équipe d'experts de notre partenaire industriel. Une plus grande variété de sources améliorerait cette étude.

7.1.9 Compte rendu

Les résultats de cette étude ont été présentés aux équipes de notre partenaire industriel, ainsi qu'à nos experts opérationnels, dans le but d'évaluer notre approche à travers ces cas d'études. Pour faire cette présentation nous avons organisé différents rendez-vous avec ces équipes. Ainsi, nous avons pu échanger autour des cas d'étude et mesurer l'intérêt de notre approche auprès des différents partenaires. La diffusion plus large de ces études a donc été décidée à travers notamment plusieurs articles scientifiques [57], [58].

7.1.10 Calendrier

La réalisation de ces cas d'études a été faite de manière itérative. Nous avons commencé par une première réunion ayant pour objectif de modéliser un seul scénario à l'aide du langage HOS-ML. Cette réunion avait pour but de mettre en évidence les éléments

nécessaires à la modélisation d'un cas d'étude, mais également de mettre en place la démarche méthodologique pour de futurs cas d'étude. À la suite de cela, nous avons identifié le fait qu'il fallait plusieurs équipes pour réaliser un cas d'étude complet. En effet l'équipe industrielle amène les éléments organisationnels et architecturaux du cas d'étude, mais les éléments liés au profil humain doivent provenir d'une autre équipe (opérationnelle). Le planning mis en place a donc été le suivant :

— Réunion avec l'équipe industrielle pour la mise au point de l'architecture du cas d'étude et des deux scénarios ;
— Réunion avec l'équipe opérationnelle pour la mise au point des profils humains ;
— Réalisation des simulations de vulnérabilité ;
— Présentation des résultats de manière individuelle à chaque expert ;
— Synthèse des différentes vulnérabilités identifiées et validées par les avis d'experts.

7.2 Étude de cas : lutte contre la piraterie maritime

Le premier cas d'étude que nous allons aborder est un exemple qui décrit le fonctionnement d'un contrôle maritime opéré par un navire dans des zones maritimes sujettes à de la piraterie maritime. Le SoSTS représenté dans cet exemple a pour objectif de venir empêcher des actes de piraterie maritime et de permettre le contrôle d'une zone géographique donnée. Cet exemple est fourni par notre partenaire industriel et il a été volontairement épuré sur certains points pour des raisons de confidentialité.

7.2.1 Contexte

La piraterie maritime est depuis longtemps un enjeu important dans le commerce international. Elle a été définie dans le traité international de Montego Bay [92]. Il s'agit d'une activité qui prend place dans les eaux internationales et qui, menée par des individus privés, a pour but de nuire à des fins d'appropriations. Cette activité est présente dans plusieurs zones géographiques qui peuvent être des zones d'intérêts stratégiques pour différents acteurs. On peut par exemple trouver dans le golfe de Guinée une piraterie très orientée autour du vol de pétrole ; au niveau du détroit de Malaga la piraterie est plutôt liée au vol sur les navires alors qu'au niveau de la Somalie, la piraterie se focalise sur de la prise d'otages. Ces différents usages montrent que la piraterie est une activité criminelle se déroulant sur des zones géographiques diverses et présentant des méthodologies radica-

lement différentes. De ce fait, un processus humain dans un SoSTS ayant pour objectif de s'adapter à chacune des situations possibles doit être naturellement robuste et résiliant à toute tentative de nuisance, cela afin de permettre une opérabilité dans tous les milieux et face à tout type de menaces.

7.2.2 Définition

Ce cas d'étude a pour objectif de décrire l'architecture socio-technique d'une chaîne fonctionnelle embarquée à bord d'un navire et visant à lutter contre des actes de piraterie maritime. Les opérateurs humains impliqués dans cette chaîne fonctionnelle vont suivre un processus de lutte qui est composé par les quatre étapes suivantes : *détecter, identifier, classifier, proposer une réponse opérationnelle*. Cette chaîne fonctionnelle est armée par cinq acteurs à bord de la passerelle du navire et qui conduisent les étapes pré-citées. Les rôles suivants décrivent ces opérateurs :

L'officier de pont : ce rôle est celui de l'officier responsable de la chaîne fonctionnelle. C'est aussi à lui que reviennent les tâches les plus critiques en termes de décision ;

Le chef de veille : ce rôle est le deuxième le plus important dans ce processus. C'est lui qui joue le rôle de pivot pour la transmission des informations ainsi que des ordres donnés ;

Le chef d'intervention : ce rôle est celui du chef d'une équipe d'intervention héliportée pouvant être projetée vers un navire extérieur ;

L'opérateur de veille : L'opérateur ayant ce rôle a comme objectif de neutraliser des navires identifiés comme hostiles ;

Le premier opérateur : Ce rôle est celui qui a la vision sur les systèmes radars et des systèmes permettant l'identification de navires.

La répartition des différentes tâches à réaliser par ces rôles, et les documents manipulés, est visible dans le tableau 7.4. Les profils humains ont été réalisés avec les données transmises par des experts opérationnels de la Marine Nationale. Les facteurs directs et indirects attachés à ces rôles sont visibles dans le tableau 7.3. Les facteurs directs utilisent des polices normales, les indirects des polices italiques. Dans cette étude de cas, certains facteurs ont la même valeur pour tous les rôles. Ceci est habituel lorsque tous les rôles du système socio-technique concernent un opérateur impliqué dans une même structure organisationnelle. Il en va autrement lorsque notre approche est appliquée à des systèmes

socio-techniques impliquant des organisations distinctes. Par exemple, si le système étudié concerne une grande entreprise mêlant à la fois ses propres employés et des employés externalisés, il serait réaliste d'imaginer que les deux types d'employés aient une différence sur les facteurs Management, Culture ou Conscience.

L'observation des profils décrits pour les rôles de cette étude de cas, notamment en termes de valeurs des propriétés des facteurs directs et indirects, permet de comprendre l'importance donnée à certains rôles du point de vue structurel. Le rôle reflétant ici le plus haut niveau d'exigence est celui d'"intervention chief" (chef d'intervention). En effet, le rôle de chef d'un commando marine requiert un niveau de qualification et d'exigence très important. A l'inverse, on peut observer que le rôle des opérateurs est moins exigeant en termes d'expérience et de compétences, et requiert pour les politiques de sécurité un niveau plus faible. Cependant, il n'en reste pas moins relativement élevé étant donné qu'aucun de ces facteurs directs n'est en dessous de 3 sur une échelle de 5. On peut attribuer ce coût minimum élevé en termes d'architecture au fait que ce sont des postes opérationnels et qui demandent donc un certain niveau de qualification.

L'architecture du SoSTS représentant ce cas d'études et décrite en HoS-ML est visible dans la figure 7.1.

La création d'une telle architecture passe par plusieurs étapes. Tout d'abord, l'architecte crée un rôle et le décrit. Par exemple, le rôle *Opérateur de surveillance* est caractérisé par ses facteurs directs et indirects. Ces derniers sont définis dans de petites cases blanches avec une étoile qui sont reliées au rôle (ou à l'acteur) par une flèche. Leurs valeurs ne sont pas visibles dans le diagramme car elles sont définies dans une fenêtre de propriété spécifique de l'outil de modélisation. Ils sont conformes à la valeur exprimée dans le tableau 7.3.

Ensuite, pour chaque rôle, l'architecte décrit les objectifs (en vert) et les documents (ou informations) que le rôle détient (en gris). Par exemple, les objectifs numéro 5, 7 et 12 sont assignés à l'opérateur de veille *Monitoring Operator*. Ces valeurs sont un raccourci vers leur définition (toutes les tâches sont disponibles dans la table 7.4). Ce rôle récupère les informations transmises par le *Système d'arme* ("Weapon System"). Lorsque tous les rôles sont définis, la collaboration entre eux doit être établie. Les cases bleues, qui sont liées aux cases vertes, désignent les délégations d'objectifs qui permettent d'illustrer la collaboration entre opérateurs ou acteurs autour d'objectifs. Les cases orange représentent les transmissions d'informations entre les rôles. Par exemple, concernant le *Opérateur de veille*, une délégation d'objectifs a été définie entre lui et le rôle Officier de pont (*Officier*) concernant l'objectif numéro 7 (appelé "Sommations" (*Summation*)). Cela signifie que le

rôle *Officer* donne une délégation d'objectifs (le 7ème objectif) à l'opérateur *Monitoring operator*. En outre, la transmission d'informations sur le système d'armes est établie entre les deux. Cela signifie que l'*Opérateur de veille* rapporte les informations du *Système d'arme* à l'officier de pont (*Officier*).

Pour la délégation d'objectifs et la transmission d'informations, l'architecte peut exprimer des contraintes de sécurité, représentées par les valeurs *confidentialité*, *intégrité* et *disponibilité*. Ces contraintes de sécurité permettent de représenter un besoin de sécurité particulier lors d'une communication fonctionnelle entre 2 acteurs.

Enfin, l'architecte crée les acteurs jouant les rôles, et fixe leurs valeurs de facteur humain. Comme mentionné, la différence entre la valeur idéale, exprimée par les rôles, et la valeur potentielle estimée, exprimée par les acteurs, introduit une vulnérabilité potentielle.

Une fois l'architecture définie, l'architecte peut dérouler des scénarios pour tester les différents profils humains affectés aux différents rôles. Pour ce faire, l'architecte génère progressivement une différence entre le profil idéal attendu (défini pour le rôle) et celui de l'opérateur. Cette opération est répétée plusieurs fois de manière itérative pour délimiter un domaine de couverture dans lequel la vulnérabilité de l'opérateur est acceptable pour l'architecte. Deux scénarios appliqués sur l'architecture de cette étude de cas sont donnés ci-après.

7.2.3 Scénario 1

Le premier scénario appliqué à ce cas d'études va mettre à l'épreuve un opérateur vis-à-vis d'un chantage qui lui est fait. Ce type d'attaque peut régulièrement apparaître aujourd'hui sur les réseaux sociaux. On les qualifie d'attaque par ingénierie sociale. Ces attaques peuvent être particulièrement dévastatrices sur la vie privée d'une personne et peuvent générer des aspects de panique chez un individu ciblé.

7.2.3.1 Contexte du scénario 1

Le navire évolue dans une zone sujette à la piraterie loin des côtes, cela fait plusieurs semaines que les marins sont en mer. Le contexte est suffisamment bon pour que les marins aient toujours accès à Internet et notamment aux réseaux sociaux.

Role / *Human Factor*	Officer	Monitor chief	Intervention chief	First operator	Monitor operator
Skill	4	4	5	3	3
Experience	4	4	5	3	3
Emotional Stability	stable	stable	unemotional	stable	stable
Reliability	4	4	4	3	3
Informational Level	4	4	5	3	3
Conscientious	efficient	efficient	efficient	efficient	efficient
Organizational Coop.	confidence	confidence	confidence	confidence	confidence
Robustness	4	4	5	4	4
Confidence	5	4	5	3	3
Culture	*simple-Structure*	*simple-Structure*	*simple-Structure*	*simple-Structure*	*simple-Structure*
Security Policy	*5*	*4*	*4*	*3*	*3*
Resource	*4*	*4*	*5*	*4*	*4*
Task Exigency	*5*	*4*	*5*	*3*	*3*
Communication	*active*	*active*	*energetic*	*active*	*active*
Management	*CB*	*CB*	*CB*	*CB*	*CB*
Position	*5*	*4*	*4*	*3*	*3*

TABLE 7.3 – Lutte contre la piraterie maritime : profils humains des rôles dans la LPM avec facteurs directs et indirects

7.2.3.2 Attaque

Le but de l'attaquant et de faire échouer la neutralisation d'une embarcation s'adonnant à de la piraterie maritime et prise en charge par le navire sur lequel le SoSTS est déployé. Pour ce faire l'attaquant doit avoir une certaine connaissance de l'organisation des marins à bord. Il choisit de cibler un opérateur de veille (*moniteur operator*). Pour cela, il utilise différentes techniques de "*phishing*" sur lui, dans lesquelles il se fera passer pour un site d'escorte en ligne par exemple. Le but est de récupérer des informations et photos compromettantes de la cible de manière à pouvoir la menacer par chantage. L'attaque se passe dans un délai relativement court, par exemple dans un délai de moins de 24 heures. L'idée est de soumettre la cible à un stress maximal. L'objectif final de l'attaquant est que, lorsque l'ordre est donné à cet opérateur de faire feu sur l'embarcation suspecte,

Task num	Task	Opered by
1	Ship labelling	First Operator
2	Report suspicious vessel	First Operator
3	Assessment of the ship's attitude and potential threat level	Monitor chief
4	Contact for identification if possible	Monitor chief
5	All civilian ships are to be placed on alert	Monitoring Operator
6	Summation order	Officer
7	Summation	Monitoring Operator
8	Prevention response team (helicopter, commandos ...)	Officer
9	Commando Intervention Order	Officer
10	Commando Intervention	Intervention chief
11	Firing order	Officer
12	Neutralization	Monitoring Operator
Document num	Document	possessed by
1	Field intelligence	Monitor chief
2	Navigation flow	First operator
3	Weapon systems	Monitoring operator

TABLE 7.4 – Tâches et documents dans les opérations de lutte contre la piraterie

il mettent plus de temps que prévu à s'exécuter de manière à ce que l'embarcation pirate puisse se mettre à l'abri.

7.2.3.3 Résultat

La simulation de ce scénario donne les éléments suivants :

— L'impact sur le système devient possible dès lors que l'on réduit l'expérience et la compétence de l'individu ainsi que sa stabilité émotionnelle, sa robustesse et sa conscience.

— L'impact est une vulnérabilité de niveau trois sur cinq avec une probabilité de 60 %. Cela pourrait correspondre à un jeune marin qui serait sous la pression d'une attaque de cyber-harcèlement.

— La propagation possible de cette vulnérabilité est visible sur la figure 7.2 en rouge. Cela signifie que, la vulnérabilité peut éventuellement toucher l'officier de pont Cette propagation de vulnérabilité pourrait ici prendre la forme par exemple d'une altération de l'information que le *Monitoring Operator* doit donner à l'officier de pont sur les systèmes d'armes. Cette altération pourrait donc à son tour rendre vulnérable l'officier.

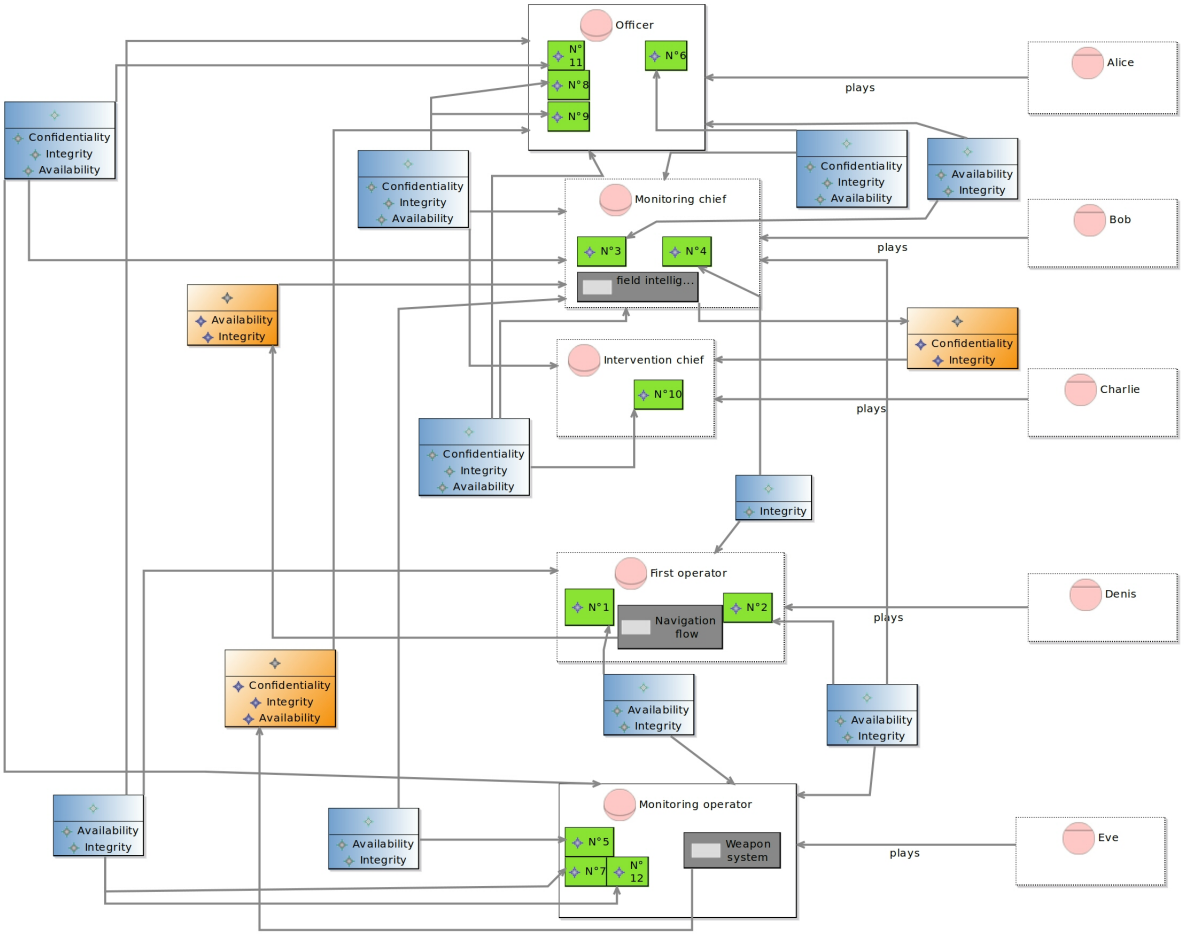


FIGURE 7.1 – Architecture du SoSTS de lutte contre la piraterie maritime exprimé avec HoS-ML

Ce niveau de vulnérabilité modéré rend possible la mise en danger du SoSTS. En effet, ici le scénario montre que l'attaque peut potentiellement réussir, suffisamment en tout cas pour atteindre les objectifs du rôle ciblé qui est celui de l'opérateur de veille (*monitor operator*) (tâche 12 : neutralisation). De plus l'attaque peut potentiellement altérer les informations transmises à un autre rôle. Cela concerne ici la communication d'informations sur les systèmes d'armes vers l'officier de pont. Cette combinaison d'évènements pourrait donc aboutir à une mise en échec du navire effectuant la lutte contre la piraterie maritime et donc conduire à la réussite d'une opération de piraterie du point de vue de l'attaquant, et cela grâce à une attaque cyber.

Les conclusions ci-dessus ont été présentées aux différents experts. L'évaluation a été menée via le questionnaire résumé par le tableau 7.1. Le tableau 7.5 montre les réponses

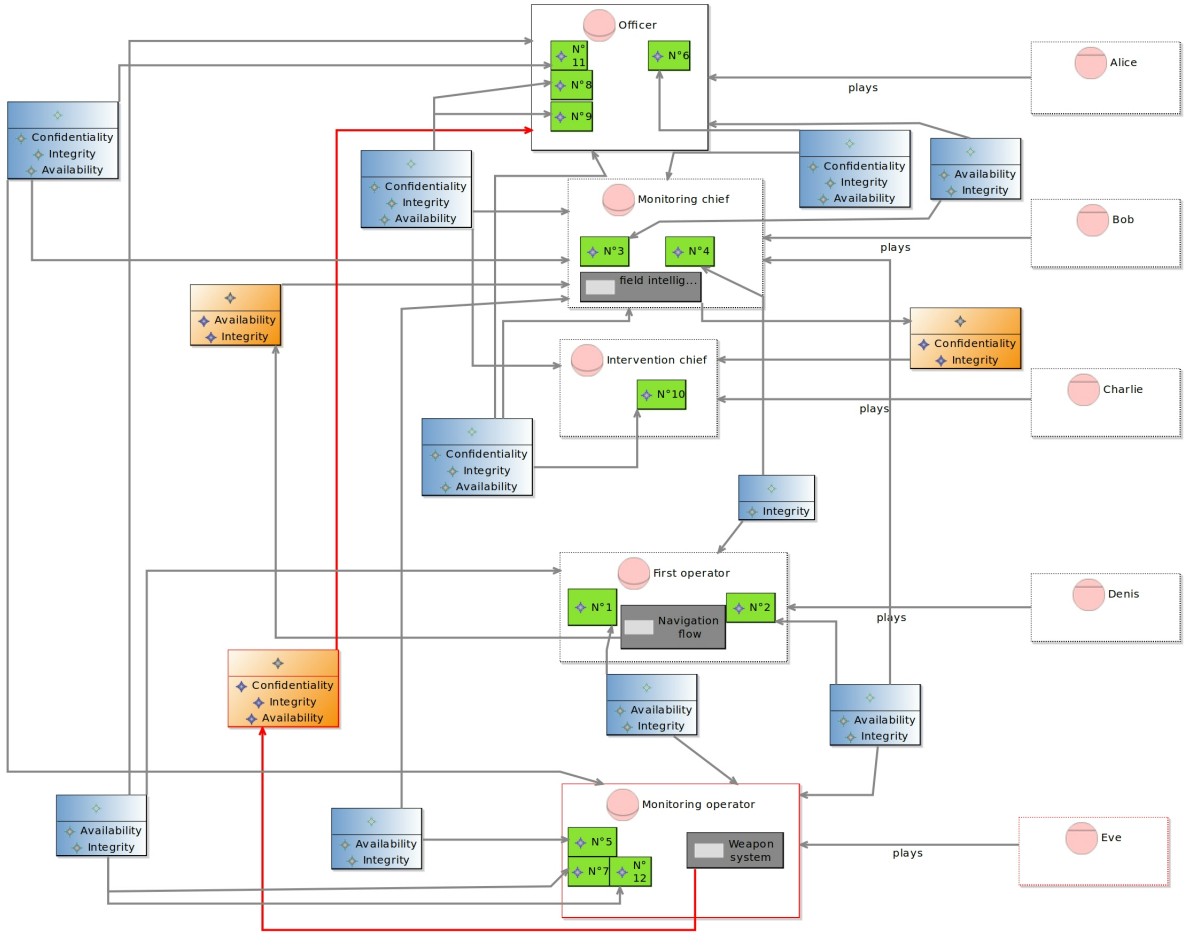


FIGURE 7.2 – Impact de la vulnérabilité humaine dans le scénario 1 du cas d'étude sur la piraterie maritime

des experts. Ici le retour des experts est globalement en accord avec le scénario et les conclusions tirées.

7.2.4 Scénario 2

Ce scénario met à l'épreuve le chef de veille (*Monitor chief*) via une attaque plus évoluée que la précédente d'un point de vue technique. Ici, les attaquants utilisent une attaque sur l'AIS [1]. L'AIS est le système anti-collision aujourd'hui utilisé par la plupart des navires civils et militaires. Il s'agit d'un système d'échanges automatisés de messages entre navires par radio VHF qui permet aux navires et aux systèmes de surveillance de

1. AIS : Système d'identification automatique ou Automatic Identification System

Question	Expert 1	Expert 2	Expert 3	Expert 4
1	Oui	Oui	Oui	Oui
2	Oui	Oui	Oui	Oui
3	4	4	4	2
4	Oui	Oui	Oui	Oui
5	Oui	Oui	Oui	Oui
6	Oui	Oui	Oui	Oui
7	Oui	Oui	Oui	Oui

TABLE 7.5 – Réponses des experts cas 1 scénario 1

trafic de connaître l'identité, le statut, la position et la route des navires se situant dans la zone de navigation. Le principal défaut de ce système est qu'il s'agit d'un système déclaratif. Ainsi, si un individu souhaite mentir, voire injecter de fausses trames, c'est possible puisqu'aucune forme de sécurité n'a été pensée lors de la création de ce standard. Ce type d'attaque a été documenté récemment à plusieurs endroits du globe [93].

7.2.4.1 Contexte du scénario 2

Pour ce scénario, le navire est dans une zone de piraterie qui est proche des côtes d'un détroit. Cette zone est une zone de très fort trafic naval avec une piraterie présente de manière récurrente. Cela pourrait être au large de la Somalie ou près du détroit de Malacca. Pour ce qui est de la durée de la mission, les marins sont en mer depuis plusieurs semaines et la fatigue de la mission commence à se faire sentir.

7.2.4.2 Attaque

Pour ce deuxième scénario, nous avons fait le choix de simuler une attaque par accommodation à un signal AIS. Le but de l'attaquant est de masquer une embarcation dans un trafic maritime. Pour ce faire, il simule une trace AIS de manière régulière, pendant une semaine par exemple. Devant une telle trace, le risque est que les marins de veille, après avoir été reconnaître plusieurs fois cette trace sans trouver d'embarcation sur zone, se rendent compte que cette trace ne reflète la présence d'aucune embarcation réelle. Ils la considèrent alors comme un défaut du système. A un moment donné, la trace sera émise à partir d'une embarcation pirate réelle. Celle dernière, bénéficiant de l'effet d'habitude des marins de surveillance, pourra alors s'approcher d'une cible. C'est l'effet dit de "Pierre et le loup".

7.2.4.3 Résultat

La simulation que nous avons menée implémentant ce scénario donne aux éléments suivants :

- Une vulnérabilité impactant de manière importante le SoSTS est identifiée. Celle-ci se produit en attaquant le rôle chef de veille (*Monitor chief*) pour lequel l'acteur jouant son rôle aura les facteurs suivants dégradés : expérience, compétence, et coopération organisationnelle, la conscience et la fiabilité ;
- La vulnérabilité obtenue est de niveaux trois sur cinq à 80%. Cette vulnérabilité demande beaucoup de facteurs ne correspondant pas à l'attendu, mais est réaliste. Elle pourrait correspondre à un jeune officier trop sûr de lui, occupant son poste depuis peu et faisant preuve de beaucoup de négligence ;
- La propagation possible de cette vulnérabilité est visible sur la figure 7.3. Elle touche 3 rôles.

Cette vulnérabilité est plus coûteuse à déclencher que la vulnérabilité précédente. Pour se produire, elle nécessite des compétences techniques relativement élevées, puisqu'elle demande un savoir-faire particulier pour générer des fausses trames AIS. Elle requiert également un écart important entre les facteurs directs du rôle et ceux de l'acteur. Il faut en effet que l'acteur baisse sa garde et qu'un nombre non négligeable des valeurs de ses facteurs directs ne correspondent pas aux valeurs attendues pour le rôle. Cela pourrait résulter d'une forme d'assurance ou d'arrogance qui pourrait être attribuée à un jeune officier prenant le poste de chef de veille (*Monitor chief*). L'impact de cette vulnérabilité est in fine sans commune mesure avec la vulnérabilité précédente. En effet, ici nous touchons pas moins de trois autres rôles dans le SoSTS et cela pourrait paralyser l'intégralité du système face à une menace qui ne sera alors pas détectée.

Les conclusions ci-dessus ont été présentées aux différents experts. L'évaluation a été menée via le questionnaire résumé par le tableau 7.1. Le tableau 7.6 montre les réponses des experts. Ici le retour des experts est globalement en accord avec le scénario et les conclusions tirées. Nous reviendrons en détail sur les réponses portées sur ce tableau puis sur les différents tableaux d'expert situés plus bas dans la partie discussion.

7.3 Étude de cas : contrôle aérien

La seconde étude de cas que nous avons menée avec notre partenaire industriel se déroule dans le domaine du contrôle aérien. Nous avons fait le choix de changer le domaine

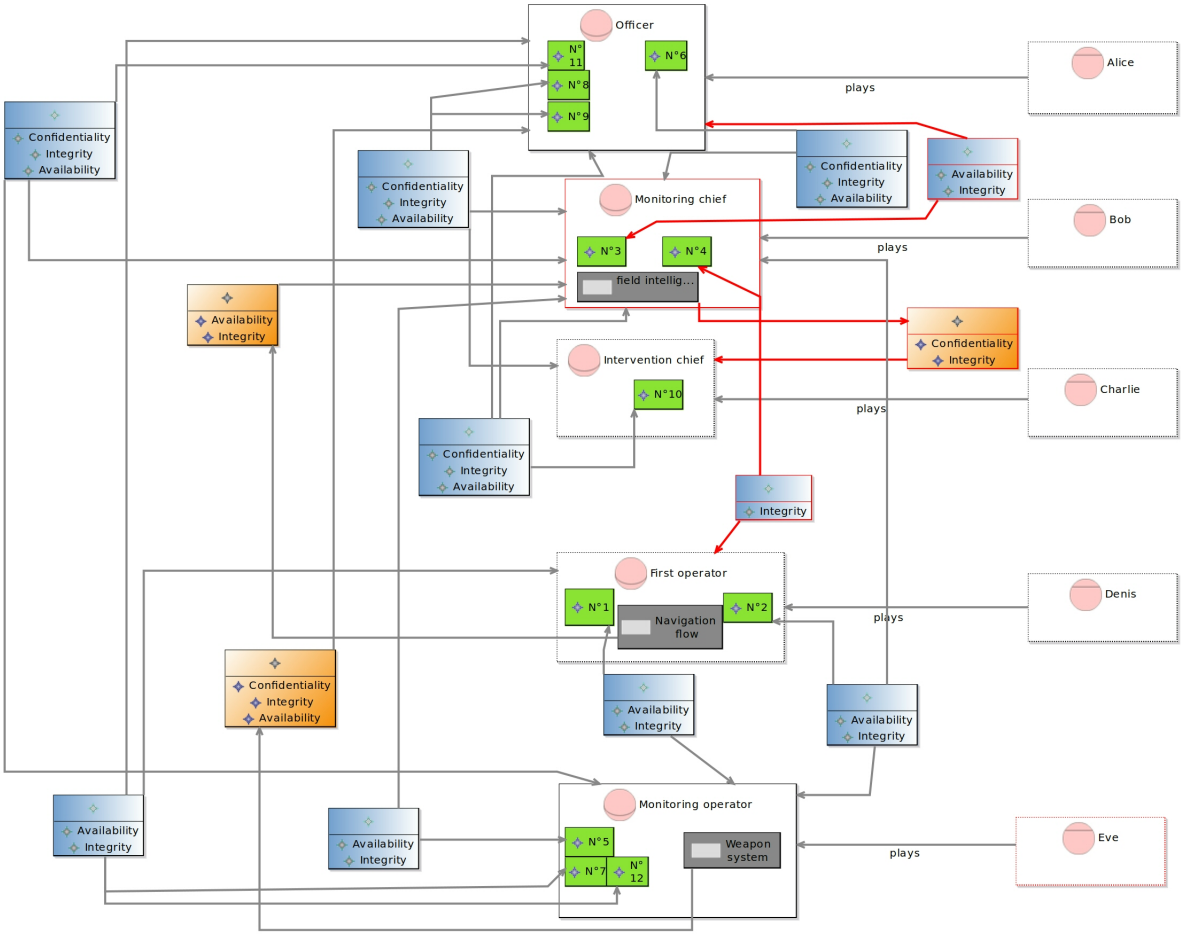


FIGURE 7.3 – Impact de la vulnérabilité humaine dans le scénario 2 du cas d'étude sur la piraterie maritime

d'application dans cette étude, de manière à montrer la non-spécificité de notre méthode au seul domaine naval et son applicabilité générique.

7.3.1 Contexte de l'étude de cas

Le secteur aéronautique est un secteur clé de nos sociétés. Aujourd'hui beaucoup de services reposent sur la capacité, qu'elle soit civile ou militaire, à faire voler des appareils. De plus, comme l'ont montré les attentats du 11 septembre 2001, le secteur aéronautique peut être détourné de manière à mener des actions malveillantes. La menace terroriste y est donc un enjeu majeur, et cela a donné lieu à différents durcissements de la sécurité et de la sûreté aérienne. Les différentes procédures et traités mis en place suite à ces

Question	Expert 1	Expert 2	Expert 3	Expert 4
1	Oui	Oui	Oui	Non
2	Oui	Oui	Oui	Non
3	2	2	5	1
4	Oui	Oui	Oui	Non
5	Oui	Oui	Oui	Oui
6	Oui	Oui	Non plus important	Oui
7	Oui	Oui	Oui	Oui

TABLE 7.6 – Réponses des experts cas 1 scénario 2

attentats ont notablement mieux régi la sécurité aérienne. D'autres événements récents impliquant la sûreté aérienne ont pu montrer la nécessité de ne pas baisser sa vigilance dans ce secteur. C'est dans ce contexte que nous avons réalisé une étude de cas dans ce domaine, et plus spécifiquement sur la gestion du trafic aérien par des opérateurs. Dans celle-ci, nous considérons une chaîne fonctionnelle de gestion du trafic aérien civil ayant pour objectif premier d'éviter aux différents appareils de subir des accidents, tant dans les airs qu'au sol.

7.3.2 Définition

L'étude de cas a pour objectif de modéliser l'interaction entre différents contrôleurs aériens intervenant à plusieurs échelons, du niveau local jusqu'au niveau régional. Le cas d'études définies cinq rôles :

Contrôleur régional (CTR) [2] : L'objectif de ce rôle est d'effectuer le contrôle de tout l'espace aérien placé sous sa responsabilité. Cela implique d'assurer la sécurité des appareils évoluant dans une zone géographique régionale dont il a la charge en appliquant les règles d'évolution liées aux appareil en vol et régis par les règles internationales.

Contrôleur de tour de contrôle (TWR) : Ce rôle a pour but de prendre en charge tous les appareils évoluant sur les pistes de l'aéroport dont il a la charge, ainsi que ceux, en vol, qui se trouvent en vue de son aéroport et qui évoluent au sein de l'espace aérien proche placé sous sa responsabilité. Il fait le lien entre plusieurs contrôleurs.

2. Ce rôle est une fusion de plusieurs rôles liée au contrôle régional. Ils ont été ici condensés pour permettre la représentation dans le cas d'études

Contrôleur d'approche (APP) : L'objet de ce rôle est de permettre la descente en toute sécurité des appareils vers une piste donnée, cela jusqu'à l'établissement de l'axe final d'approche de la piste. C'est aussi ce rôle qui donne les autorisations pour l'atterrissage des appareils en approche.

Contrôleur de départ (DEP) : Ce rôle a pour but de gérer les appareils qui sont sur le départ et d'attribuer les autorisations pour le décollage.

Contrôleur sol (GND) : Ce contrôleur est spécialisé sur le guidage des appareils au sol. Plus spécifiquement, il a pour objectif de gérer tant au roulage qu'a l'arrêt les appareils au sol ainsi que leur répartition sur l'aéroport, cela notamment pendant qu'ils sortent ou arrivent sur une piste.

La définition précise des objectifs de chaque rôle ainsi que des documents et des informations qu'ils manipulent sont visibles dans le tableau 7.7. Les profils humains des différents opérateurs sont décrits dans le tableau 7.8. Nous pouvons noter que les profils humains correspondant aux différents rôles présentent tous un fort niveau d'exigence. Cela est dû au haut degré de fiabilité attendu d'un contrôleur aérien. En effet, pour tous, le niveau de qualification requis est important et cela est retranscrit sur les facteurs directs et indirects des rôles pris en compte. L'un des rôles de contrôleur aérien est d'ailleurs jugé comme particulièrement critique. Il s'agit du contrôleur d'approche. Il a en effet un niveau d'exigence particulière élevé et cela se traduit par la valeur du facteur humain gérant le statut émotionnel qui est spécifiée à "*unemotional*". Son comportement ne doit en aucun cas être guidé par ses émotions. Il a en effet un rôle crucial pour éviter des accidents lors de la phase d'approche qui est l'une des plus critiques pour les avions de ligne ; il doit donc avoir un sang-froid à toute épreuve. Le but de ce cas d'étude est donc de modéliser le SoSTS ainsi décrit et d'évaluer par simulation si une attaque cyber peut impacter une telle organisation.

Le déroulement de la modélisation avec HoS-ML suit le même processus que pour le cas d'étude portant sur la piraterie maritime et décrit dans la section 7.2.2. La figure 7.4 montre le modèle réalisée avec HoS-ML et notre outil.

7.3.3 Scénario 1

Le premier scénario envisagé dans cette étude de cas se place dans un contexte où un individu souhaiterait perturber le trafic aérien d'un aéroport. Pour ce faire, l'attaquant va essayer de perturber la communication entre les contrôleurs et les avions qui se déroule en

Task num	Task	Opered by
1	Ensure the control service of the aircrafts	CTR
2	Ensure aircraft safety by managing flight levels	CTR
3	Provide pre-regulation for approaches using radar guidance	CTR
4	Give approach clearance to aircraft	APP
5	Ensure the safety of the devices in the approach	APP
6	Ensure separation of departures	DEP
7	Ensure that pilots can continuously climb as much as possible to reach the control en route	DEP
8	Give directions or radar guidance to perform the above tasks or shorten trajectories.	DEP
9	Give landing, take-off and runway crossing authorizations	TWR
10	Control the airspace near the field	TWR
11	Manage aircraft movements on the ground only in the parking and taxiway areas.	GND
12	Order the traffic on the ground	GND
13	Coordinate with the TWR controller if an aircraft needs to cross a runway to reach the appropriate stopping point	GND
Document num	Document	Possessed by
1	Situation information	CTR
2	Flight information	GND

TABLE 7.7 – Tâches et documents dans les opérations de contrôle aérien

VHF. Pour cela, un attaquant peut utiliser une attaque dite de l'"homme du milieu" [94]. Il s'agit d'une attaque ayant pour objectif d'intercepter les communications entre deux parties, sans que ni l'une ni l'autre puisse se douter que le canal de communication entre elles a été compromis. Ce type d'attaque peut avoir de lourdes conséquences et générer un arrêt complet du trafic aérien autour de l'aéroport, mais peut aussi mettre en périls des avions ayant peu de réserve de carburant.

7.3.3.1 Contexte

Dans ce scénario, un appareil souhaite se poser sur un aérodrome. Il est pris en charge normalement par le contrôleur régional et le suivi de l'appareil est ensuite donné au contrôleur d'approche. L'avion est un long-courrier qui vient se poser après un long vol, avec un niveau de carburant faible mais sans être pour autant critique.

Role / *Human Factor*	CTR	DEP	APP	TWR	GND
Skill	5	4	5	4	3
Experience	5	4	5	4	3
Emotional Stability	stable	stable	unemotional	stable	stable
Reliability	4	4	4	4	3
Informational Level	4	4	4	3	3
Conscientious	efficient	efficient	efficient	efficient	efficient
Organizational Coop.	confidence	confidence	confidence	confidence	confidence
Robustness	5	4	5	4	3
Confidence	5	4	5	4	3
Culture	*simple Structure*	*simple Structure*	*simple Structure*	*simple Structure*	*simple Structure*
Security Policy	*4*	*4*	*4*	*4*	*4*
Resource	*5*	*4*	*4*	*4*	*4*
Task Exigency	*5*	*4*	*5*	*4*	*3*
Communication	*active*	*active*	*active*	*active*	*active*
Management	*CB*	*CB*	*CB*	*CB*	*CB*
Position	*5*	*4*	*4*	*4*	*3*

TABLE 7.8 – Facteurs directs et indirects des profils humains entrant en compte dans la gestion du contrôle aérien

7.3.3.2 Attaque

Le but de l'attaque est de perturber le contrôleur d'approche au moyen de la VHF. Pour ce faire l'attaquant va essayer de perturber et d'empêcher au maximum l'approche de l'avion en direction de la piste en envoyant par VHF des informations et ordres contradictoires à ceux transmis par le contrôleur d'approche.

7.3.3.3 Résultat

La simulation opérée donne les éléments suivants :

— Le rôle du contrôleur d'approche est suffisamment critique pour venir impacter les objectifs du SoSTS. Il suffit donc qu'il devienne faiblement vulnérable pour que cela présente un risque significatif. Pour ce faire, l'acteur jouant ce rôle se voit

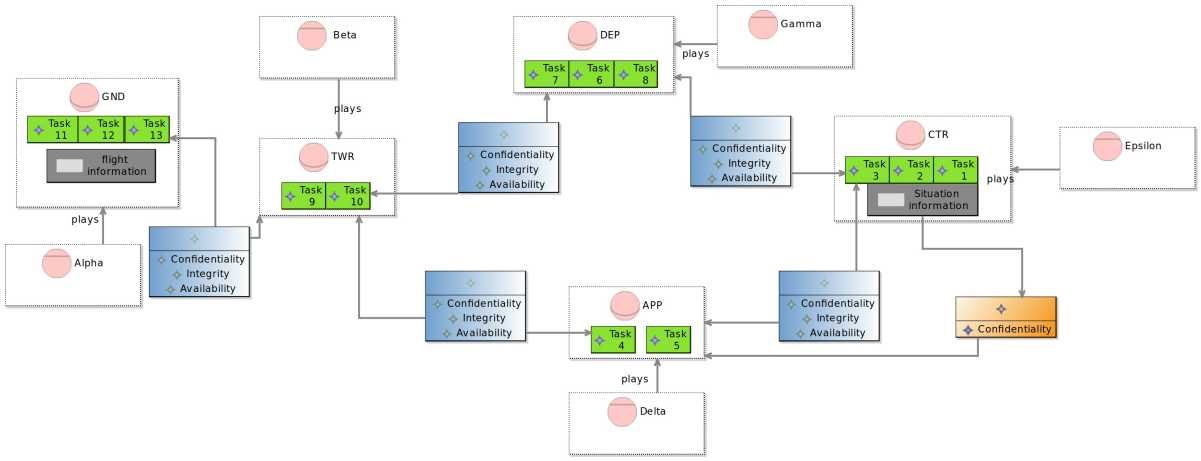


FIGURE 7.4 – Architecture du SoSTS du contrôle aérien en utilisant HoS-ML

dégrader légèrement les facteurs suivants : fiabilité, compétence et expérience. La simulation donne une vulnérabilité de niveau 2 sur 5 à 60 % de probabilité.

— L'impact que peut avoir le contrôleur d'approche reste malgré son niveau de vulnérabilité contenue, car en effet, il n'atteint pas, dans cette configuration, d'autres contrôleurs du SoSTS.

— La propagation possible de cette vulnérabilité est visible sur la figure 7.5. Cette propagation montre que la vulnérabilité ne touche que le contrôleur d'approche.

Cette vulnérabilité montre qu'avec un opérateur présentant un déficit de formation relativement faible ou quelques lacunes légères, il est possible d'avoir un impact important sur le système. La vulnérabilité technique de la VHF permet d'atteindre directement l'opérateur. En cas d'une telle attaque, il y a de fortes chances pour que cette attaque réussisse et que la vulnérabilité identifiée conduise à empêcher l'atterrissage d'un appareil. Cette vulnérabilité pourrait même aller jusqu'à générer suffisamment de troubles et amener ainsi un certain nombre d'appareils à se retrouver avec des réserves de carburant critiques.

Les conclusions ci-dessus ont été présentées aux différents experts avec le questionnaire 7.1. Le tableau 7.9 montre les réponses des experts à ces questions. Les différentes réponses des experts par rapport à ce scénario montrent qu'ils sont en accord avec les conclusions obtenues.Nous reviendrons en détail sur les réponses de ce tableau et des différents tableaux d'expert dans la partie discussion située plus bas.

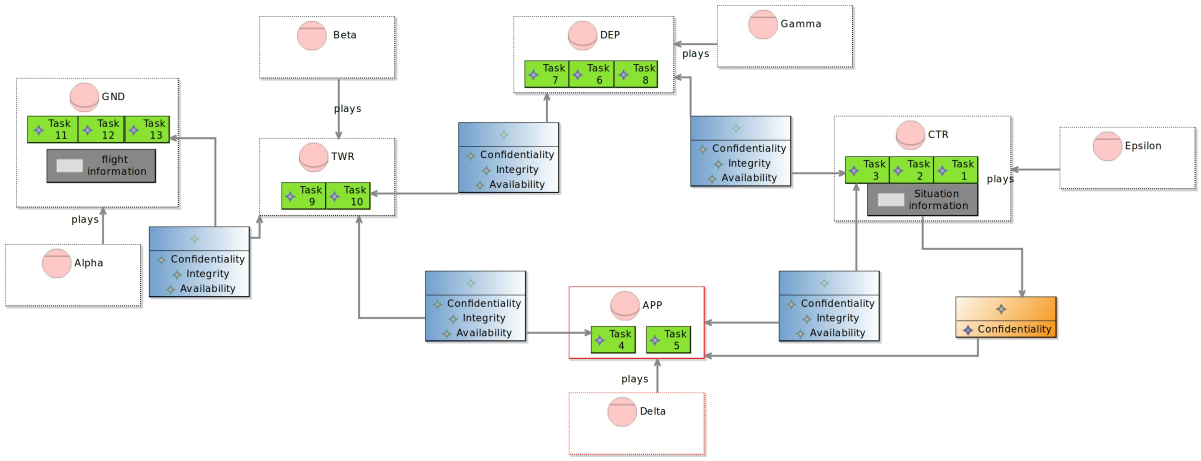


FIGURE 7.5 – Impact de la vulnérabilité humaine dans le scénario 1 du l'étude de cas sur le contrôle aérien

Question	Expert 1	Expert 2	Expert 3	Expert 4
1	Oui	Oui	Oui	Oui
2	Non	Oui	Oui	Oui
3	1	2	3	2
4	Oui	Oui	Oui	Oui
5	Oui	Oui	Oui	Oui
6	Oui	Oui	Oui	Oui
7	Oui	Oui	Oui	Oui

TABLE 7.9 – Réponses des experts scénario 1 de l'étude de cas portant sur le contrôle aérien

7.3.4 Scénario 2

Le deuxième scénario lié au contrôle aérien est un scénario qui ressemble beaucoup par son attaque technique au second scénario du cas d'étude portant sur la piraterie maritime. En effet ici le vecteur d'attaque est l'ADS-B [3] qui est un protocole anti-collision semblable à l'AIS pour le secteur aérien.

7.3.4.1 Contexte

Dans ce scénario, le contexte est celui d'un aéroport d'importance régionale accueillant des vols internationaux. L'attention de ce scénario ne se situe pas sur une action précise du SoSTS représenté, mais sûr la gestion quotidienne de l'intégralité des flux entrants et

3. ADS-B - Automatic Dependent Surveillance-Broadcast

sortants par les différents contrôleurs de l'aéroport.

7.3.4.2 Attaque

Dans ce scénario l'attaquant va utiliser la technologie liée à l'ADS-B pour mettre en œuvre son attaque [95]. L'attaquant va essayer de créer une accoutumance de même nature que celle du deuxième scénario de la piraterie maritime. Ici la cible de cette attaque est le rôle contrôleur régional. L'attaque lancera tous les jours un avion fictif sur des trajectoires pouvant percuter d'autres avions. Cette trame ADS-B a toujours la même trajectoire et les mêmes identifiants. Le contrôleur est tenté à un moment donné de l'ignorer pour faciliter l'approche des avions des aéroports. Le but de l'attaquant est à terme de placer un véritable aéronef de petite taille sous cette piste ADS-B, cela de manière à provoquer une collision avec un avion devant décoller ou se poser sur cet aéroport.

7.3.4.3 Résultat

La simulation donne les éléments suivants :

Pour obtenir une vulnérabilité importante sur le rôle de contrôleur régional et impacter ainsi le SoSTS, il faut arriver à la situation suivantes :

- Pour obtenir une vulnérabilité de niveau 3 sur 5 avec 60 % de probabilité, les valeurs des facteurs suivants ont été réduites : l'expérience, la compétence, la fiabilité, le niveau informationnel ainsi que la stabilité émotionnelle qui est passée à "anxieux".
- Cela représente une personne qui n'est pas très à l'aise dans son travail et qui malgré une compétence attendue élevée, n'a pas tout à fait l'expérience du métier. Avec ces éléments réunis, le système peut être suffisamment impacté pour être mis en échec.
- La propagation possible de cette vulnérabilité est visible sur la figure 7.6. Elle montre que le contrôleur régional peut contaminer au moins deux autres personnes, que ce soit le contrôleur de départ ou le contrôleur approche. Cette contamination pourrait ici s'expliquer par exemple par de mauvaises indications qu'il donnerait dans les tâches qu'il doit effectuer en collaboration avec le contrôleur de départ et le contrôleur d'approche. En effet, le contrôleur régional pourrait utiliser les informations de l'ADS-B pour l'aider à faire la régulation des différents avions et donc donner de mauvaises consignes aux autres contrôleurs les rendant ainsi vulnérables.

Dans ce scénario, la vulnérabilité identifiée est, de la même manière que pour la vulnérabilité AIS, possible mais coûteuses à obtenir en termes techniques. Cependant, elle est moins coûteuses à obtenir d'un point de vue humain. En effet, dans le scénario lié à la piraterie maritime et à l'AIS, la possibilité de rendre la vulnérabilité possible présupposait de faire occuper le poste décrit par le rôle par un jeune officier (alors qu'était attendu un officier expérimenté) ce qui était peu probable. Ici il suffit simplement que l'individu ne soit pas tout à fait bien formé, nerveux de nature et un peu négligent dans son travail pour permettre à cette attaque de réussir. Ce scénario nous semble d'autant plus probable que des évènements similaires sont déjà arrivés et cela sans même une attaque cyber intentionnelle [4].

Les conclusions ci-dessus ont été présentées aux différents experts avec le questionnaire 7.1. Le tableau 7.10 montre la réponse des experts à ces questions. Les experts ici se montrent en accord avec les conclusions obtenus et jugent d'ailleurs la probabilité de cet événement comme étant assez forte.

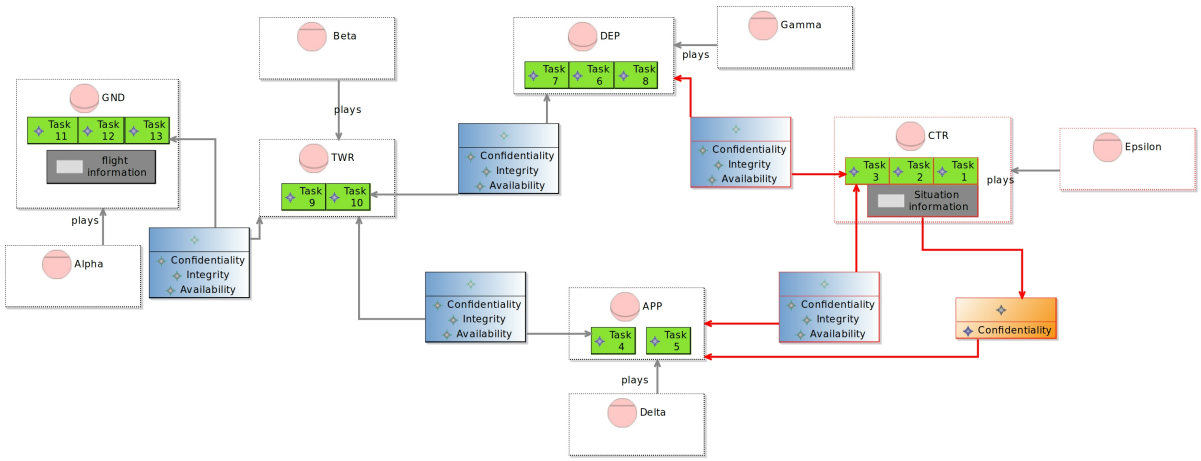


FIGURE 7.6 – Impact de la vulnérabilité humaine dans le scénario 2 de l'étude de cas sur le contrôle aérien

7.4 Discussion sur les résultats des études de cas

Les deux études de cas que nous avons menées dans le cadre de ces travaux de recherche, et que nous vous avons présentées dans ce chapitre, nous ont permis de mettre

4. https ://www.theguardian.com/business/2007/may/16/theairlineindustry.travel

Question	Expert 1	Expert 2	Expert 3	Expert 4
1	Oui	Oui	Oui	Oui
2	Oui	Oui	Oui	Oui
3	4	4	5	3
4	Oui	Oui	Oui	Oui
5	Oui	Oui	Oui	Oui
6	Oui	Oui	Oui	Oui
7	Oui	Oui	Oui	Oui

TABLE 7.10 – Réponses des experts au scénario 2 de l'étude de cas sur le contrôle aérien

à l'épreuve à la fois, nos modèles visant à caractériser les humains (vis-à-vis des risques cyber), notre langage, ainsi que l'outil de simulation que nous avons réalisé.

Les résultats que nous avons obtenus, au travers de la mise en œuvre de cette méthodologie de validation, nous permettent de discuter des questions de recherche que nous avons mises en place au préalable à cette étude. Concernant la question QR1C, nous avons mis à l'épreuve notre approche dans la réalisation de quatre situations décrites au moyen de deux études de cas et pour chacune deux scénarios. Nous avons ainsi, lors de la réalisation de ces études, et en lien avec l'équipe d'architectes, évalué chacune des étapes de notre méthodologie ainsi que le langage et les outils. Nous avons pu discuter de la pertinence et des limites de l'approche à propos de la phase de spécification des rôles des acteurs dans un SoSTS et lors de la mise en forme des architectures étudiées avec notre langage.

Les résultats mettent en évidence la pertinence de la méthode utilisée et la capacité à représenter les SoSTS avec notre langage. Le tableau 7.11 résume l'avis des experts. Dans les limites identifiées, il y a le passage à l'échelle. En effet, les experts ont noté que ce langage et notre outils formaient une preuve de concept convaincante. Cependant, sa forme actuelle ne permet pas des représentations de SoSTS complexes car cela générerait une surcharge visuelle rendant difficile la manipulation du système représenté. Elle peut toutefois être prise en compte et réduite en utilisant l'abstraction ou le découpage afin de manipuler des modèles de taille raisonnable.

Pour ce qui est de QR2C, les résultats obtenus comprennent les tableaux allant de 7.5 à 7.10. Ces tableaux nous permettent d'apprécier le retour des experts sur la pertinence des résultats fournis par notre langage et notre outil sur les scénarios réalisés. Il est cependant à noter que, pour le deuxième scénario de l'étude de cas sur la piraterie maritime (attaque AIS), un des experts a émis des doutes sur la probabilité et la réelle représentation

Question	Expert 3	Expert 4
1	Oui	Oui
2	Oui	Oui
3	Oui	Oui
4	Oui	Oui
5	Oui seulement pour représenté de petit SoSTS	Oui

TABLE 7.11 – Réponses des experts sur le langage HoS-ML

d'une telle attaque. L'expert a expliqué ce point de vue par l'existence d'une contre-mesure sur les frégates de nouvelle génération. Cette contre-mesure nous était inconnue au moment de la réalisation du scénario. Elle permet malgré tout de valider la pertinence du résultat, puisque la vulnérabilité identifiée par notre approche l'avait également été précédemment par une équipe d'experts industriels, entraînant sa prise en compte au moyen d'un correctif.

Outre la validation générale de notre approche et de ses actuelles limites identifiées par les experts, il convient de noter que notre approche se concentre exclusivement sur la partie humaine des SoSTS. Afin de développer une plus grande capacité de détection et une prise en compte plus fine de la vulnérabilité, il faudrait ajouter à notre approche la prise en compte de la partie technique du SoSTS, permettant ainsi une combinaison des éléments à la fois humains et techniques. De plus, concernant les probabilités manipulées par le réseau bayésien, il conviendra d'adapter leurs valeurs à la spécificité des domaines métiers traités. En effet, nos valeurs actuelles, bien qu'étant issues de la littérature et d'expertises de notre partenaire industriel, ne sont pas adaptées à la prise en compte de n'importe quel domaine métier. Ainsi, plus les architectes utilisateurs de notre approche auront la possibilité de raffiner et spécifier les valeurs des probabilités en utilisant des données métiers, plus la simulation sera performante.

De manière plus générale, et au regard des éléments donnés par les experts, nous pouvons conclure que les cas d'études ont permis de montrer une réelle pertinence de ces travaux sur l'analyse de la vulnérabilité humaine dans les SoSTS. Le formalisme retenu pour estimer la vulnérabilité humaine présente des limitations, mais est déjà suffisamment puissant pour permettre une adaptabilité et un apprentissage sur de nouvelles données. Ainsi, il permettra de mieux évaluer la vulnérabilité en fonction des domaines ainsi que des choix opérés par l'architecte. Enfin, il reste une marge d'amélioration autour de la représentation et de l'interfaçage des systèmes techniques avec les systèmes humains pris en compte.

7.4.1 Résumé du chapitre

Dans ce chapitre, nous avons confronté nos différentes contributions à des études de cas industrielles. Cette confrontation apporte un premier niveau de validation et permet de dresser les premières limites que nous pouvons tracer sur nos différentes contributions.

Dans un premier temps nous avons présenté la méthodologie des cas d'études que nous avons mise en place. Nous exposons les différentes étapes de cette méthodologie qui vise à présenter le contexte dans lequel ont été menés ces cas d'études et toutes les étapes par lesquelles nous sommes passées au cours de l'étude pour arriver à l'expérimentation.

Le premier cas d'étude réalisée se concentre sur la lutte contre la piraterie maritime. Après avoir défini le contexte de ce cas d'étude, nous avons décrit les deux scénarios qui en sont tirés. Le premier scénario soumet un des opérateurs de cette architecture à une attaque via des réseaux sociaux. Le deuxième scénario va soumettre l'architecture à une attaque plus sophistiquée visant un système technique de bord qui est AIS, en utilisant des informations fallacieuses que transmettra ce système aux opérateurs. Dans chacun de ces scénarios, le résultat de la simulation a été soumis à des experts du domaine en architecture et en cybersécurité.

Le deuxième cas d'étude réalisé se concentre sur le contrôle aérien dans un petit aéroport. Après avoir défini le contexte de ce cas d'étude, nous avons décrit deux scénarios qui se déroulent dessus. Le premier est une attaque portant sur l'architecture définie en utilisant une tactique simple dite "l'homme du milieu". Le deuxième scénario va tester la résilience de l'architecture sur une attaque portant sur le système ADS-B. Dans chacun de ces scénarios, le résultat de la simulation a été lui aussi soumis à des experts du domaine en architecture et en cybersécurité.

Ce chapitre se termine par une discussion sur les résultats des cas d'études. En effet, les experts du domaine ont été confronté à notre méthodologie d'estimation de la vulnérabilité humaine dans un SoSTS. Aussi, leurs commentaires nous ont permis de déterminer certaines limites notamment liées au langage. Ils ont également émis des réticences raisonnables sur la probabilité des événements sur certains scénarios. Cependant, l'ensemble des retours est globalement très positif.

Quatrième partie

Conclusion et Perspectives

Chapitre 8

Conclusion

Sommaire

8.1 Problématique

Lors de la conception d'un système de systèmes socio-technique (SoSTS), la prise en compte de la vulnérabilité humaine est un enjeu primordial pour permettre une bonne sécurisation du système. En effet, les SoSTS, en plus de permettre de nouveaux usages, mettent en évidence la prépondérance de l'utilisateur et donc celle du facteur humain en termes de vulnérabilités potentielles.

Dans ce document, nous avons proposé d'enrichir la conception sécurisée des systèmes en intégrant la prise en compte de la vulnérabilité humaine dans les SoSTS. L'objectif de cette démarche est de permettre une amélioration du niveau de sécurité du système en développement via une meilleure détection en amont des vulnérabilités pouvant potentiellement le mettre en défaut.

Pour ce faire nous avons identifié trois questions de recherche et une question de développement :

— Un besoin méthodologie et de représentation des SoSTS (QR1).
— Un besoin de modélisation du facteur humain (QR2).
— Un besoin de simulation et d'estimation de cette vulnérabilité (QR3).
— Un besoin d'outils de mise en œuvre (QD1).

8.2 Travaux réalisés

Pour répondre de manière appropriée à notre problématique qui est l'analyse de la vulnérabilité humaine dans les systèmes de systèmes socio-techniques, il est nécessaire dans un premier temps de comprendre ce qu'est un système socio-technique et comment on peut construire un tel système. Pour ce faire, nous avons commencé par réaliser une étude de contexte et un état de l'art présentant ce que sont les systèmes de systèmes socio-techniques. Au cours de ce travail, nous avons également étudier la manière de concevoir de tels systèmes. Nous avons ensuite focalisé notre attention sur la définition de la notion de cybersécurité, puis de vulnérabilité humaine et enfin nous nous sommes intéressés à l'estimation de cette dernière. Nous avons enfin regardé la manière d'agréger ces concepts afin de concevoir de manière sécurisée un tel système. C'est avec cette vision du contexte ainsi que de l'état de l'art que nous avons pu poser nos questions de recherches.

Par la suite, nos travaux se sont portés en premier lieu sur la QR2. Cela nous a amené à définir la notion de facteur humain dans les SoSTS. Nous avons pu établir que celle-ci était liée à la description de l'humain et de son environnement qui peut-être faite à l'aide de facteurs directs et indirects. Les facteurs directs ont pour but de représenter l'individu et les facteurs indirects l'environnement. Les différents facteurs que nous avons retenus sont extraits de la littérature venant des sciences humaines. Nous avons ajouté à ces facteurs une gradation pour permettre la réalisation d'un profil humain pouvant être évalué informatiquement. Nous en avons dérivé un métamodèle représentation l'humain dans notre contexte. Celui-ci utilise les facteurs directs et indirects identifiés dans ce document.

Pour répondre à la QR1, nous nous sommes basés sur un langage existant et permettant de représenter les systèmes socio-techniques (ST). Ce langage a comme avantage de permettre de représenter un certain nombre d'éléments organisationnels et architecturaux. Nous avons extrait de ce langage plusieurs représentations et les avons mêlées à nos éléments de représentation liées issus de notre métamodèle humain. Ce nouveau langage a été nommé HoS-ML. Nous avons pu lui associer une méthodologie d'usage, que nous avons définie, permettant de sécuriser la construction des SoSTS. Dans cette méthodologie l'objectif est de comparer un acteur effectif à l'acteur idéal qui est défini par l'architecte du système. La différence entre l'acteur effectif et l'acteur idéal d'un rôle va permettre d'estimer la vulnérabilité humaine et son impact.

Pour répondre à la QR3 nous avons mis en place une estimation de la vulnérabilité

humaine utilisant des données venant de la littérature liée aux sciences humaines et sociales. Pour permettre la manipulation de ces données et faire des conjectures avec celles-ci vis-à-vis d'une architecture que l'on souhaite simuler, nous avons utilisé l'approche probabiliste. Cette approche est basée sur un réseau bayésien qui permet de venir lier notre modèle de facteur humain avec l'architecture, le tout en permettant la prise en compte de données de la littérature. Cette approche a aussi l'avantage de permettre l'apprentissage, ainsi, si un utilisateur de cette simulation le souhaite, il peut venir enrichir le réseau avec ses propres données.

Pour mettre en œuvre les solutions que nous proposons à travers les trois questions de recherche, nous avons développé un outil, nommé HoS-ML Editor, à l'aide d'Obeo designer. Cet outil permet l'utilisation du langage et de la méthodologie qui lui est liée, ainsi que la mise en œuvre de simulations. Cet outil permet la manipulation de notre langage qui vient faciliter l'usage de la simulation sur la vulnérabilité humaine que nous avons pu décrire.

Enfin, nous avons confronté nos différentes contributions à deux cas d'étude que nous avons mis en place avec notre partenaire industriel. Le premier cas d'étude est celui d'un équipage à bord d'une frégate impliquée dans la lutte contre la piraterie maritime. Le deuxième cas d'usage est celui d'une tour de contrôle d'un aéroport devant contrôler le trafic aérien passant dans sa zone de contrôle aérien ainsi que différents avions décollant et atterrissant. Ces deux cas d'études ont été modélisés en utilisant notre méthodologie et notre langage. Nous avons par la suite procédé à des simulations sur l'architecture permettant de mettre en évidence plusieurs vulnérabilités potentielles. Ces vulnérabilités potentielles ont ensuite été présentées à des experts en architecture et en cybersécurité, ce qui nous a permis de venir mettre à l'épreuve et valider nos contributions. Cette validation s'est faite en utilisant une méthodologie de validation issue de la littérature.

Le travail que nous avons pu réaliser à travers cette thèse représente une première étape vers la résolution de la problématique du traitement de la vulnérabilité humaine dans les systèmes de systèmes socio-techniques.

8.3 Discussion

Nos contributions sur la prise en compte de la vulnérabilité humaine dans les SoSTS visent à répondre aux problèmes qui sont associés à nos questions de recherche initialement définies. De ce fait, les contributions présentées ont un certain nombre de limitations, que

ce soit au niveau méthodologique, que sur leur application au travers des cas d'études. Dans cette section nous souhaitons présenter de manière ouverte les différentes limitations de nos travaux.

Le modèle de facteur humain développé pour répondre à la QR2 cherche à être le plus générique possible au vu de la vulnérabilité humaine et de ses multiples facettes. Ce modèle reste particulièrement adapté au contexte des SoSTS. Cependant, la vulnérabilité humaine dans le contexte de la cybersécurité ne s'applique pas qu'aux SoSTS. Pourtant notre modèle de facteur humain n'est pas forcément adapté à tous les contextes et toutes les typologies de systèmes différents. Ce modèle peut avoir ignoré des facteurs qui sont spécifiques à certains scénarios ou certains besoins qui n'ont pas été identifiés. En conséquence, il peut être nécessaire d'adapter ce modèle de facteur humain aux spécificités de l'architecture et de l'ingénierie système où il sera employé.

La représentation de l'architecture d'un SoSTS, en adéquation avec notre QR1 présente plusieurs limitations. Premièrement même si nous restons dans un cadre de SoSTS, nous n'avons pas réellement intégré les systèmes techniques dans la représentation d'architecture. Les systèmes techniques sont cependant une partie très important de la définition des SoSTS et leur absence fait qu'il manque un certain nombre d'éléments dans la prise en compte de la vulnérabilité des SoSTS. Nous avons proposé la modélisation du facteur humain dans ce contexte, mais sans prendre en compte la vulnérabilité existante dans les systèmes techniques, ni la combinaison de ces deux éléments. Les systèmes techniques peuvent aussi rentrer en ligne de compte lors des attaques à rebond, et leur absence dans la représentation choisie ne permet pas des scénarios complexes d'attaque visant un SoSTS.

La solution que nous avons choisie pour répondre à la QR3 s'appuie sur le concept de réseau bayésien. La principale limitation de cette approche est celle des données qui viennent nourrir ce réseau. Un réseau bayésien peut en effet apprendre à partir de données métier mais si ces données ne sont pas disponibles, il faut alors une solution de repli. N'ayant à ce jour que très peu de données venant de la littérature et sans données métier, l'approche que nous avons prise avec ce réseau a été de lui fournir des données ad-hoc dans l'attente de pouvoir les remplacer par des données venant de la littérature ou du métier. Ces données, même si elles s'inspirent de règles que nous avons pu extraire de la littérature, n'en restent pas moins que des données ad-hoc et donc ne représentent pas fidèlement la réalité. Pour compenser en partie cela, nous avons mis en place une méthode de pondération permettant à un expert de venir modifier les nœuds dans le réseau. Cette pondération modifie le résultat du réseau bayésien en fonction de ce que l'expert peut

penser *a priori* face à l'architecture. Cette méthode pour venir compenser le manque de données réelles dans le réseau bayésien est discutable. L'expertise d'un individu ne permettant pas toujours d'obtenir les réglages de pondération idéaux pour un système et un contexte donné.

L'outil informatique que nous avons développé lié à la QD1, permet de faciliter l'usage et de démontrer la possibilité d'implémentation des trois questions de recherche précédentes, mais il présente à ce jour certaines limitations dans son utilisation. Premièrement, ce n'est pas un outil finalisé, mais bien un démonstrateur, il présente donc des erreurs résiduelles qui peuvent nuire à l'expérience utilisateur. Deuxièmement, son usage pour réaliser la conception d'un système de systèmes socio-techniques complet présente plusieurs difficultés. L'une des principales est l'absence de possibilité de venir scinder l'architecture en plusieurs vues ce qui rend rapidement les architectures complexes à lire et à maintenir. Une autre difficulté est liée à la méthode de retour qui est faite à l'utilisateur suite à une simulation. Celui-ci est en effet restreint à un simple affichage graphique venant donner les conclusions de la simulation. Sur plusieurs utilisations successives, un rapport de synthèse aurait permis une meilleure restitution à l'utilisateur.

Enfin, pour ce qui est de la preuve de concept que nous avons réalisée sur deux cas d'études composés de deux scénarios, nous pouvons noter comme limitation particulière le fait que notre équipe a réalisé les architectures des cas d'études et cela même si les données d'architecture venaient de notre partenaire industriel. Cela peut introduire un biais sur l'architecture modélisée puisqu'elle n'a pas été réalisée par des architectes de métier, mais seulement supervisée par eux. De plus il faudra réaliser beaucoup plus de cas d'études pour tester les limites liées aux différentes contributions. En effet, les deux cas d'études ont une architecture assez simple et ils ne permettent pas de traiter une complexité à grande échelle, comme par exemple celle d'un navire, et de venir estimer toutes les implications qu'une vulnérabilité peut introduire.

Chapitre 9

Perspectives

Le sujet étudié dans cette thèse a nécessité l'exploration de plusieurs domaines, allant de l'analyse du comportement humain à l'ingénierie système, en passant par la gestion du management et de la cybersécurité. La pluralité de ces domaines nous a permis de soulever un certain nombre de questions et de limites, lesquelles ont pu notamment être exposées dans la discussion précédente (8.3). Les réponses apportées dans cette thèse à toutes ces problématiques ne sont que le sommet émergent d'un iceberg gigantesque.

Le travail réalisé pour répondre à QR1 a montré qu'il pourrait être intéressant d'approfondir la conception de SoSTS. Par exemple, il serait essentiel de pouvoir intégrer les systèmes techniques aux propositions de la thèse en intégrant la représentation de ces systèmes dans le langage HoS-ML. Leur intégration permettrait de venir prendre en compte toute la partie vulnérabilité technique qui reste actuellement manquante dans le langage. Cela permettrait à terme une recherche sur la fusion d'un système de systèmes socio-techniques avec la prise en compte simultanée de la vulnérabilité humaine et de la vulnérabilité technique pour ainsi permettre d'étudier l'émergence de vulnérabilités lorsque ces deux éléments sont présents. On pourrait intégrer à cela les arbres d'attaque qui permettraient de représenter un schéma souvent complexe d'attaque sur les différents éléments composant le SoSTS. En complément, il pourrait être intéressant de venir explorer la conjugaison de ces différentes approches de représentation des SoSTS avec l'évolution du système existant. Car pour le moment l'approche que nous avons choisie a été de se baser sur une conception a priori d'un système et non pas d'appliquer cette approche sur des systèmes déjà existants et permettant de venir potentiellement corriger des vulnérabilités existantes post-conception.

Parmi les perspectives que nous ont amenées les éléments de réponse que nous avons pu développer pour la QR2, se dégage notamment la possibilité d'élargir notre méthode d'estimation de la vulnérabilité humaine en y intégrant toute la partie assignation de l'information à un opérateur humain. Cet ajout permettrait de venir prendre en compte le type d'attaque informationnelle opérée sur l'individu et le niveau d'adhésion potentielle

que celui-ci pourrait avoir en fonction de sa capacité à croire ou non à l'information. Pour permettre de prendre en compte la capacité d'un individu à croire dans une information, il faudrait pour cela se pencher sur les biais cognitifs [96]. Cela permettrait aussi de venir décrire plus finement la capacité d'une attaque à être plus efficace sur certains profils d'individus. Une autre perspective serait de transférer notre méthode d'estimation de la vulnérabilité humaine dans d'autres contextes que celui des SoSTS. Les SoSTS sont en effet paradigmes particuliers où cette vulnérabilité existe, mais ce ne sont pas les seuls. Venir adapter nos propositions pour d'autres paradigmes permettrait de mieux prendre en compte le facteur humain dès la conception. En effet, des systèmes plus spécifiques peuvent avoir des besoins liés à la vulnérabilité humaine comme pour les "cyber-physical system" (CPS). Adapter le modèle de manière plus contextuelle pour des besoins spécifiques peut avoir un intérêt sur la précision de l'estimation de la vulnérabilité humaine ainsi que pour des éléments de conception sécurisée à mettre en place.

L'approche développée pour répondre aux différentes problématiques posées avec QR3, nous permet également de dégager plusieurs perspectives. La première perspective est liée aux jeux de données qui alimentent le réseau bayésien. Il faudrait mettre en place, pour les humains les procédures de collecte de données existantes pour les systèmes techniques [97]. Ces données une fois collectées pourraient permettre de venir simuler de manière beaucoup plus précise la vulnérabilité humaine. Une autre possibilité pour avoir de tels jeux de données serait au travers de diverses enquêtes et études, sur les incidents liés au facteur humain. Une deuxième perspective concerne la matrice de pondération. Pour obtenir une simulation à partir d'un réseau bayésien beaucoup plus significative, il pourrait être intéressant de mettre en place des pré-réglages de matrice de pondération, par domaine et par typologie de SoSTS.

Le développement de l'outil correspondant à QD1 permet d'automatiser un certain nombre d'éléments posés par les trois questions de recherche. Malgré cela, l'outil pourrait être plus poussé pour permettre un meilleur accompagnement de l'utilisateur dans sa quête de conception sécurisée. L'une des approches qui pourraient aider l'utilisateur serait celle des patrons de conception [98]-[100]. En effet, une base de connaissances pourrait être constituée et développée à partir de modèles d'architecture existants, sécurisés et adaptés, dans le contexte des SoSTS et pourrait être directement proposée de manière sécurisée à l'utilisateur. Pourrait être également développée l'aide à la mise en place de profils humains pouvant correspondre au rôle décrit par l'architecte. Il est difficile pour un architecte dont l'expertise ne porte pas sur la conception de profils humains de venir

caractériser les profils devant occuper idéalement des rôles. Ici l'enjeu serait de venir créer des profils types en fonction des niveaux de compétences pouvant être attendus vis-à-vis du rôle ainsi que du type de fonctions que pourrait remplir un individu, permettant ainsi plus facilement à l'architecte utilisateur de venir faire correspondre des profils idéaux au rôle.

Enfin, pour ce qui est de la validation, une étude de cas d'architectures beaucoup plus complexes pourrait permettre de grandement améliorer les différents apports. Même si les deux cas d'étude que nous avons formulées sont des situations basées sur des données industrielles et réelles, elles ont été réalisées dans un environnement expérimental et n'ont pas été menées de bout en bout dans le cadre d'un processus d'ingénierie système. La perspective serait ici de mener de bout en bout une conception d'un système complexe comprenant plusieurs dizaines d'acteurs et de systèmes techniques.

Liste de publications

1. Paul Perrotin, Salah Sadou, David Hairion, Antoine Beugnard. *Detecting human vulnerably in socio-technical systems : a naval case study.*, 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, MODELS-C 2020 - Companion Proceedings

2. Paul Perrotin, Nicolas Belloir, Salah Sadou, David Hairion, Antoine Beugnard *Hos-ML : Socio-Technical System ADL Dedicated to Human Vulnerability Identification* , *International conference* : 26th International Conference on Engineering of Complex Computer Systems (ICECCS 2022)

3. Paul Perrotin, Nicolas Belloir, Salah Sadou, David Hairion, Antoine Beugnard. *Using the architecture of Socio-Technical System to analyse its vulnerability*
217th Annual System of Systems Engineering Conference, (SOSE 2022)

Acronymes

ADS-B Automatic Dependent Surveillance-Broadcast.

AFIS Association Française d'Ingénierie Système.

AIS Automatic Identification System.

ANSSI Agence Nationale de la Sécurité des Systèmes d'Information.

CPS Cyber-Physical System.

DNN Deep Neural Network.

EBIOS Expression des Besoins et Identification des Objectifs de Sécurité.

EMF Eclipse Modeling Framework.

GMF Graphical Modeling Framework.

HoS-ML Human-oriented Security architecture Modeling Language.

IHM Interface Homme-Machine.

INCOSE International Council on Systems Engineering.

IoT Internet of Things.

ISO International Organization for Standardization.

MDE Model-Driven Engineering.

NASA National Aeronautics and Space Administration.

NIST National Institute of Standards and Technology.

SDLC Security Development Life Cycles.

SoS System of System.

SoSTS Socio-Technical System of Systems.

ST Socio-Technical system.

STS-ML Socio-Technical Security Modeling Language.

SWOT Strengths Weaknesses Opportunities Threats.

SysML Systems Modeling Language.

UIT Union Internationale des Télécommunications.

BIBLIOGRAPHIE

[1] P. S. Christopher Davey Paul Nielsen, *SYSTEMS ENGINEERING vision 2035 ENGINEERING SOLUTIONS FOR A BETTER WORLD*, 2021.

[2] Verizon, « Data Breach Investigations Report », Verizon, rapp. tech., 2016.

[3] ARCEP, « BAROMÈTRE DU NUMÉRIQUE Edition 2021 », ARCEP, rapp. tech., 2021.

[4] A.-M. IDRAC, « DÉVELOPPEMENT DES VÉHICULES AUTONOMES Orientations stratégiques pour l'action publique », Ministère de la Transition écologique et de la Cohésion des territoires, rapp. tech., 2018.

[5] M. D. Vicario, W. Quattrociocchi, A. Scala et F. Zollo, « Polarization and Fake News : Early Warning of Potential Misinformation Targets », *ACM Trans. Web*, 2019.

[6] V. U. et A. Gandhi P, « Prediction of COVID-19 Outbreaks Using Google Trends in India : A Retrospective Analysis », *Healthcare Informatics Research*, 2020.

[7] « ISO/IEC/IEEE International Standard - Systems and software engineering – System life cycle processes », *ISO/IEC/IEEE 15288 First edition 2015-05-15*, p. 1-118, 2015. doi : 10.1109/IEEESTD.2015.7106435.

[8] R. C. (in Chief), *The Guide to the Systems Engineering Body of Knowledge (SEBoK), v.(2.6)*. SEBoK Editorial Board, 2022.

[9] M. W. Maier, « Architecting principles for systems-of-systems », *Systems Engineering : The Journal of the International Council on Systems Engineering*, t. 1, *4*, p. 267-284, 1998.

[10] K. E. Boulding, « General systems theory—the skeleton of science », *Management science*, t. 2, *3*, p. 197-208, 1956.

[11] C. B. Nielsen, P. G. Larsen, J. Fitzgerald, J. Woodcock et J. Peleska, « Systems of Systems Engineering : Basic Concepts, Model-Based Techniques, and Research Directions », *ACM Comput. Surv.*, t. 48, *2*, 18 :1-18 :41, sept. 2015, issn : 0360-0300. doi : 10.1145/2794381. adresse : http://doi.acm.org/10.1145/2794381.

[12] I. Engineering, *INCOSE Systems Engineering Handbook : A Guide for System Life Cycle Processes and Activities*. juin 2015, isbn : 9781118999400.

[13] A. Association Française d'Ingénierie Système, « Découvrir et comprendre l'Ingénierie Système », p. 259, 2009.

[14] R. Cooper et M. Foster, « Sociotechnical systems. », *American Psychologist*, t. 26, *5*, p. 467-474, 1971. doi : 10.1037/h0031539. adresse : https://doi.org/10.1037/h0031539.

[15] B. Pavard, J. Dugdale et P. Salembier, « Conception de systèmes sociotechniques robustes », *La securité en action. Toulouse : Octarès*, avr. 2009.

[16] R. Lock et I. Sommerville, « Modelling and Analysis of Socio-Technical System of Systems », in *Proceedings of the 2010 15th IEEE International Conference on Engineering of Complex Computer Systems*, 2010.

[17] D. Sales et L. Buss Becker, « Systematic Literature Review of System Engineering Design Methods », in *2018 VIII Brazilian Symposium on Computing Systems Engineering (SBESC)*, p. 213-218. doi : 10.1109/SBESC.2018.00040.

[18] C. Haskins, K. Forsberg, M. Krueger, D. Walden et D. Hamelin, « Systems engineering handbook », in *INCOSE*, International Council on Systems Engineering Seattle, t. 9, 2006, p. 13-16.

[19] S. J. Kapurch, *NASA systems engineering handbook*. Diane Publishing, 2010.

[20] N. B. Ruparelia, « Software development lifecycle models », *ACM SIGSOFT Software Engineering Notes*, t. 35, *3*, p. 8-13, 2010.

[21] K. Forsberg et H. Mooz, « The relationship of system engineering to the project cycle », t. 1, *1*, p. 57-65, 1991.

[22] J. Dahmann et G. Roedler, « SYSTEMS OF SYSTEMS ENGINEERING STANDARDS », *INSIGHT*, t. 19, *3*, p. 23-26, 2016. doi : https://doi.org/10.1002/inst.12102. eprint : https://onlinelibrary.wiley.com/doi/pdf/10.1002/inst.12102. adresse : https://onlinelibrary.wiley.com/doi/abs/10.1002/inst.12102.

[23] T. Nguyen, « A modelling & simulation based engineering approach for socio-cyber-physical systems », in *14th IEEE International Conference on Networking, Sensing and Control, ICNSC 2017, Calabria, Italy, May 16-18, 2017*, 2017.

[24] J. G. Turnley, A. Wachtel, K. Munoz-Ramos, M. J. Hoffman, J. H. Gauthier, A. Speed et R. S. Kittinger, « Modeling Human-Technology Interaction as a Sociotechnical System of Systems. », mai 2017. doi : 10.1109/SYSOSE.2017.7994934.

[25] J. Chigada et R. Madzinga, « Cyberattacks and threats during COVID-19 : A systematic literature review », *South African Journal of Information Management*, t. 23, *1*, p. 1-11, 2021.

[26] « SÉRIE X : Réseaux de données, communication entre systèmes ouverts et sécurité », Union internationale des télécommunications, rapp. tech., 2008, Recommendation X.1205. adresse : https://www.itu.int/rec/T-REC-X.1205-200804-I.

[27] « Information technology - Security techniques - Information security management systems - Overview and vocabulary », ISO, rapp. tech., 2018, standard ISO/IEC 27000 :2018. adresse : https://www.iso.org/standard/73906.html.

[28] R. Ross, M. McEvilley et J. Oren, « Systems security engineering : Considerations for a multidisciplinary approach in the engineering of trustworthy secure systems », National Institute of Standards et Technology, rapp. tech., 2016.

[29] R. Kissel, K. M. Stine, M. A. Scholl, H. Rossman, J. Fahlsing et J. Gulick, *Sp 800-64 rev. 2. security considerations in the system development life cycle*, 2008.

[30] M. Mori, A. Ceccarelli, P. Lollini, B. Frömel, F. Brancati et A. Bondavalli, « Systems-of-systems modeling using a comprehensive viewpoint-based SysML profile », *Journal of Software : Evolution and Process*, 2018.

[31] M. Mohsin, Z. Anwar, G. Husari, E. Al-Shaer et M. A. Rahman, « IoTSAT : A formal framework for security analysis of the internet of things (IoT) », in *2016 IEEE Conference on Communications and Network Security (CNS)*, 2016.

[32] J. E. Hachem, « A Model Driven Method to Design and Analyze Secure Systems-of-Systems Architectures. Application to Predict Cascading Attacks in Smart Buildings », 2017.

[33] R. Benabidallah, I. Cherfa, S. Sadou et M. A. Nacer, « Situation/Reaction Paradigm for SoS Simulation », in *2017 IEEE 26th International Conference on Enabling Technologies : Infrastructure for Collaborative Enterprises (WETICE)*, 2017.

[34] N. Li, J. Cámara, D. Garlan et B. Schmerl, « Reasoning about When to Provide Explanation for Human-involved Self-Adaptive Systems », p. 195-204, 2020. doi : 10.1109/ACSOS49614.2020.00042.

[35] S. Trösterer, E. Beck, F. Dalpiaz, E. Paja, P. Giorgini et M. Tscheligi, « Formative user-centered evaluation of security modeling : Results from a case study », *International Journal of Secure Software Engineering*, t. 3, p. 1-19, jan. 2012.

[36] C. Boletsis, R. Halvorsrud, J. Pickering, S. Phillips et M. Surridge, « Cybersecurity for SMEs : Introducing the Human Element into Socio-technical Cybersecurity Risk Assessment », in *VISIGRAPP (3 : IVAPP)*, jan. 2021, p. 266-274. doi : 10.5220/0010332902660274.

[37] A. Chapanis, W. R. Garner et C. T. Morgan, « Applied experimental psychology : Human factors in engineering design. », 1949.

[38] S. B. Kraemer, « An Adversarial Viewpoint of Human and Organizational Factors in Computer and Information Security », AAI3234846, thèse de doct., University of Wisconsin at Madison, USA, 2006, isbn : 9780542888250.

[39] M. S. Sanders et E. J. McCormick, « Human factors in engineering and design », *Industrial Robot : An International Journal*, t. 25, *2*, 1998.

[40] K. J. Knapp, T. E. Marshall, R. Kelly Rainer et F. Nelson Ford, « Information security : management's effect on culture and policy », *Information Management & Computer Security*, t. 14, *1*, p. 24-36, jan. 2006, issn : 0968-5227. doi : 10.1108/09685220610648355. adresse : https://doi.org/10.1108/09685220610648355.

[41] S. Furnell et K.-L. Thomson, « From culture to disobedience : Recognising the varying user acceptance of IT security », *Computer Fraud & Security*, t. 2009, *2*, p. 5-10, 2009, issn : 1361-3723. doi : https://doi.org/10.1016/S1361-3723(09)70019-3. adresse : https://www.sciencedirect.com/science/article/pii/S1361372309700193.

[42] H. W. Glaspie et W. Karwowski, « Human factors in information security culture : A literature review », in *International Conference on Applied Human Factors and Ergonomics*, Springer, 2017, p. 269-280.

[43] R. Mortazavi-Alavi, « A Risk-Driven Investment Model for Analysing Human Factors in Information Security », thèse de doct., University of East London, 2016.

[44] D. W. Pickton et S. Wright, « What's swot in strategic analysis ? », *Strategic change*, t. 7, *2*, p. 101-109, 1998.

[45] L. S. Ferro, A. Marrella et T. Catarci, « A Human Factor Approach to Threat Modeling », in *International Conference on Human-Computer Interaction*, Springer, 2021, p. 139-157.

[46] A. Shostack, *Threat modeling : Designing for security*. John Wiley & Sons, 2014.

[47] A. AlHogail, « Design and validation of information security culture framework », *Computers in human behavior*, t. 49, p. 567-575, 2015.

[48] S. H. Bakry, « Development of security policies for private networks », *International Journal of Network Management*, t. 13, *3*, p. 203-210, 2003.

[49] A. N. de la Sécurité des Systèmes d'Information (ANSSI), *EBIOS Risk Manager*, 2019.

[50] K.-K. R. Choo et A. Dehghantanha, *Handbook of Big Data Privacy*. Springer, 2020.

[51] J. Kim, N. Shin, S. Jo et S. Kim, « Method of intrusion detection using deep neural network », in *2017 IEEE international conference on big data and smart computing (BigComp)*, 2017. adresse : https://www.scopus.com/inward/record.uri?eid=2-s2.0-85017650589&doi=10.1109%2fBIGCOMP.2017.7881684&partnerID=40&md5=074eaf3a3cb2072c8af490fe3f0eaafb.

[52] N. Moustafa et J. Slay, « UNSW-NB15 : a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set) », in *2015 Military Communications and Information Systems Conference (MilCIS)*, 2015, p. 1-6. DOI : 10.1109/MilCIS.2015.7348942.

[53] P. Laso, D. Brosset et J. Puentes, « Dataset of Anomalies and Malicious Acts in a Cyber-Physical Subsystem », *Data in Brief*, t. 14, juill. 2017. DOI : 10.1016/j.dib.2017.07.038.

[54] N. Koroniotis, N. Moustafa, E. Sitnikova et B. Turnbull, « Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics : Bot-IoT dataset », *Future Generation Computer Systems*, 2019. adresse : https://www.scopus.com/inward/record.uri?eid=2-s2.0-85066442910&doi=10.1016%2fj.future.2019.05.041&partnerID=40&md5=9dd27b3877ff3d963a8951a57036af9c.

[55] N. Feng, H. J. Wang et M. Li, « A security risk analysis model for information systems : Causal relationships of risk factors and vulnerability propagation analysis », *Information sciences*, t. 256, p. 57-73, 2014.

[56] S. Zhang, C. Li, L. Zhang, M. Peng, L. Zhan et Q. Xu, « Quantification of human vulnerability to earthquake-induced landslides using Bayesian network », *Engineering Geology*, t. 265, p. 105 436, 2020.

[57] P. Perrotin, S. Sadou, D. Hairion et A. Beugnard, « Detecting Human Vulnerably in Socio-Technical Systems : A Naval Case Study », in *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems : Companion Proceedings*, 2020.

[58] P. Perrotin, N. Belloir, S. Sadou, D. Hairion et A. Beugnard, « Hos-ML : Socio-Technical System ADL Dedicated to Human Vulnerability Identification », in *26th International Conference on Engineering of Complex Computer Systems (ICECCS) 2022, Hiroshima City, Japan, March*, 2022.

[59] S. Barsade, « The ripple effect : Emotional contagion and its influence on group behavior », 2002.

[60] M. Schumacher, E. Fernandez, D. Hybertson et F. Buschmann, *Security Patterns : Integrating Security and Systems Engineering*. John Wiley & Sons, Inc., 2005.

[61] R. Vogel, « Closing the cybersecurity skills gap », *Salus Journal*, t. 4, *2*, p. 32-46, 2016.

[62] S. Parkin, A. van Moorsel et R. Coles, « An Information Security Ontology Incorporating Human-Behavioral Implications », in *Proc. of the 2nd International Conference on Security of Information and Networks*, 2009.

[63] A. Avizienis, J. .-. Laprie, B. Randell et C. Landwehr, « Basic concepts and taxonomy of dependable and secure computing », *IEEE Transactions on Dependable and Secure Computing*, 2004.

[64] K.-L. Thomson et J. van Niekerk, « Combating information security apathy by encouraging prosocial organisational behaviour », *Information Management & Computer Security*, 2012.

[65] C. Vroom, « Towards information security behavioral compliance », *Computers & Security*, 2004.

[66] J.-M. Hoc et C. Chauvin, « COOPERATIVE IMPLICATIONS OF THE ALLOCATION OF FUNCTIONS TO HUMANS AND MACHINES », *MISSING*, 2011.

[67] S. Bosworth, W. O. L. (en ligne), M. Kabay et E. Whyne, *Computer Security Handbook*. Wiley, 2015.

[68] V. S. Rossouw, « Information security management : why standards are important », *Information Management Computer Security*, 1999.

[69] H. F. Tipton et M. Krause, *Information Security Management Handbook, Volume 1*. Auerbach Publications, 2007.

[70] A. Ruighaver, S. Maynard et S. Chang, « Organisational security culture : Extending the end-user perspective », *Computers & Security*, 2007.

[71] G. Dhillon et J. Backhouse, « Technical Opinion : Information System Security Management in the New Millennium », *Journal of Health Psychology*, 2000.

[72] M. A. Quinones, J. K. Ford et M. S. Teachout, « THE RELATIONSHIP BETWEEN WORK EXPERIENCE AND JOB PERFORMANCE : A CONCEPTUAL AND META-ANALYTIC REVIEW », *Personnel Psychology*, 1995.

[73] M. Al-Awadi, « Success factors in information security implementation in organizations », 2008.

[74] G. H. Hofstede, *Cultures and organizations : Software of the mind*. McGraw-Hill, 1991.

[75] J. Klink, R. Blonk, A. Schene et F. Dijk, « The benefit of interventions for work related stress », *American journal of public health*, 2001.

[76] E. Paja, F. Dalpiaz et P. Giorgini, « Modelling and Reasoning About Security Requirements in Socio-technical Systems », *Data Knowl. Eng.*, 2015.

[77] P. Perrotin, N. Belloir, S. Sadou, D. Hairion et A. Beugnard, « Using the architecture of Socio-Technical System to analyse its vulnerability », in *2022 17th Annual System of Systems Engineering Conference (SOSE)*, 2022, p. 361-366. doi : 10.1109/SOSE55472.2022.9812648.

[78] J. Pearl, « Chapter 3 - MARKOV AND BAYESIAN NETWORKS : Two Graphical Representations of Probabilistic Knowledge », in *Probabilistic Reasoning in Intelligent Systems*, J. Pearl, éd., San Francisco (CA) : Morgan Kaufmann, 1988, p. 77-141, isbn : 978-0-08-051489-5. doi : https://doi.org/10.1016/B978-0-08-051489-5.50009-6. adresse : https://www.sciencedirect.com/science/article/pii/B9780080514895500096.

[79] X.-S. Yang, « 2 - Mathematical foundations », in *Introduction to Algorithms for Data Mining and Machine Learning*, X.-S. Yang, éd., Academic Press, 2019, p. 19-43, isbn : 978-0-12-817216-2. doi : https://doi.org/10.1016/B978-0-12-817216-2.00009-0. adresse : https://www.sciencedirect.com/science/article/pii/B9780128172162000090.

[80] M. H. Janne, A. Eirik et H. Jan, « Implementation and effectiveness of organizational information security measures, policy », *Information Management & Computer Security*, 2008.

[81] J. Chatman et S. Barsade, « Personality, Organizational Culture and Cooperation : Evidence From a Business Simulation », *Administrative Science Quarterly*, 1995.

[82] H. Nouri et R. Parker, « The relationship between budget participation and job performance : The roles of budget adequacy and organizational commitment », *Accounting, Organizations and Society*, 1998.

[83] I. Bragard, A.-M. Etienne, I. Merckaert, Y. Libert et D. Razavi, « Efficacy of a Communication and Stress Management Training on Medical Residents' Self-efficacy, Stress to Communicate and Burnout : A Randomized Controlled Study », *Journal of Health Psychology*, 2010.

[84] J. A. Veltman et A. W. K. Gaillard, « Physiological workload reactions to increasing levels of task difficulty », *Ergonomics*, 1998.

[85] N. Bolger, A. DeLongis, R. Kessler et E. Wetherington, « The Contagion of Stress Across Multiple Roles », *J Marriage Fam*, 1989.

[86] S. Goel, A. Anderson, J. Hofman et D. J. Watts, « The Structural Virality of Online Diffusion », *Management Science*, 2016.

[87] M. Moussaïd, M. Kapadia, T. Thrash, R. W. Sumner, M. Gross, D. Helbing et C. Hölscher, « Crowd behaviour during high-stress evacuations in an immersive virtual environment », *Journal of The Royal Society Interface*, 2016.

[88] P. Brereton, B. Kitchenham, D. Budgen et Z. Li, « Using a protocol template for case study planning », *Proceedings of EASE*, t. 2008, jan. 2008.

[89] R. K. Yin, *Case study research : Design and methods*. sage, 2009, t. 5.

[90] N. Jones et B. Tivnan, « Cyber Risk Metrics Survey, Assessment and Implementation Plan », MITRE CORP BEDFORD MA, rapp. tech., 2018.

[91] S. C. Payne, « A Guide to Security Metrics », in *ISANS Security Essentials GSEC Practical Assignment, Ver. 1.2e*, 2006.

[92] U. Nations, « United Nations Convention on the Law of the Sea », United Nations, Official convention 31363, 1982. adresse : `https://www.un.org/depts/los/convention_agreements/texts/unclos/unclos_f.pdf`.

[93] A. Androjna, T. Brcko, I. Pavic et H. Greidanus, « Assessing Cyber Challenges of Maritime Navigation », *Journal of Marine Science and Engineering*, t. 8, 2020.

[94] R. Zhang, G. Liu, J.-y. Liu et J. Nees, « Analysis of Message Attacks in Aviation Data-Link Communication », *IEEE Access*, 2017.

[95] M. Schäfer, V. Lenders et I. Martinovic, « Experimental Analysis of Attacks on Next Generation Air Traffic Communication », in *Applied Cryptography and Network Security*, M. Jacobson, M. Locasto, P. Mohassel et R. Safavi-Naini, éd., 2013, p. 253-271.

[96] S. Raponi, Z. Khalifa, G. Oligeri et R. Di Pietro, « Fake News Propagation : A Review of Epidemic Models, Datasets, and Insights », t. 16, *3*, sept. 2022, issn : 1559-1131. doi : 10.1145/3522756. adresse : https://doi.org/10.1145/3522756.

[97] H. Hindy, D. Brosset, E. Bayne, A. K. Seeam, C. Tachtatzis, R. Atkinson et X. Bellekens, « A Taxonomy of Network Threats and the Effect of Current Datasets on Intrusion Detection Systems », *IEEE Access*, 2020.

[98] E. Fernandez-Buglioni, *Security patterns in practice : designing secure architectures using software patterns*. John Wiley & Sons, 2013.

[99] C. Dougherty, K. Sayre, R. C. Seacord, D. Svoboda et K. Togashi, « Secure design patterns », CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, rapp. tech., 2009.

[100] M. Schumacher, E. Fernandez-Buglioni, D. Hybertson, F. Buschmann et P. Sommerlad, *Security Patterns : Integrating security and systems engineering*. John Wiley & Sons, 2013.

DOCTORAT BRETAGNE LOIRE / MATHSTIC

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Titre : Analyse de la vulnérabilité humaine dans les systèmes de systèmes socio-techniques

Mot clés : Cybersécurité – Vulnérabilités humaine - Conception sécurisée

Résumé : De nos jours, la construction de systèmes se fait majoritairement via l'interconnexion de systèmes déjà existants dans lesquels l'humain tient une telle place qu'il est lui-même considéré comme un système. On parle alors de système de systèmes socio-technique (SoSTS). En ingénierie, l'un des principaux problèmes actuels posés est celui de la conception sécurisée de ce type de systèmes. En effet, l'humain est aujourd'hui le principal vecteur utilisé par les attaquant cybernétiques. Dans cette thèse, nous proposons une méthodologie permettant la détection de la vulnérabilité humaine au sein d'une architecture d'un SoSTS. Plus particulièrement, nous avons spécifié un métamodèle permettant de capturer les facteurs humains ayant un impact potentiel sur la vulnérabilité humaine dans un SoSTS. Pour cela, nous proposons un langage, nommé HoS-ML, permettant de modéliser une architecture d'opérateurs humains composant une chaîne fonctionnelle dans un SoSTS. A partir des modèles produits avec ce langage, nous avons défini une méthode d'estimation de la vulnérabilité humaine et de ses impacts sur le SoSTS. Celle-ci s'appuie sur une approche probabiliste basée sur un réseau bayésien. Enfin, nous avons implémenté cette approche, et le langage associé, dans un logiciel appelé HoS-ML Editor. Nous avons confrontés notre approche face à des cas d'études industriels et nous avons soumis les résultats à des experts industriels. Ceci nous a permis de tester, discuter et estimer les limites de notre approche.

Title: Analysis of human vulnerability in socio-technical systems of systems

Keywords: Cybersecurity - Human vulnerabilities - Secure design

Abstract: Nowadays, the construction of systems is mostly done via the interconnection of already existing systems in which the human being holds such a place that she/he is considered as an already existing system too. This is called a system of socio-technical systems (SoSTS). In engineering, one of the main current problems is the secure design of this type of systems. Indeed, nowadays the human is the main vector used by cybernetic attackers. In this thesis, we propose a methodology allowing the detection of the human vulnerability within an architecture of a SoSTS. More specifically, we have specified a metamodel to capture human factors that have a potential impact on human vulnerability within a SoSTS. For that aim, we-propose a language, called HoSML, in order to model an architecture of human operators composing a functional chain of a SoSTS. Based on the models produced with that language, we have defined a method for estimating human vulnerability and and its impacts on the SoSTS. This method is based on a probabilistic approach, nemelly Bayesian network. Finally, we have implemented this approach, and the associated language, as a software called HoS-ML Editor. We have validated our approach using industrial case studies and we have submitted the results to industrial experts.This allowed us to test, discuss and estimate the limits of our approach.