Un voyage dans les modèles : du logiciel aux systèmes de systèmes socio-techniques

Nicolas Belloir

HABILITATION A DIRIGER DES RECHERCHES DE

L'UNIVERSITE
DE BRETAGNE SUD

ECOLE DOCTORALE N° 644
Mathématiques et Sciences et Technologies
de l'Information et de la Communication en Bretagne Océane
Spécialité : *Informatique et Architectures Numériques*

Par

« Nicolas BELLOIR »

« Un voyage dans les modèles : du logiciel aux systèmes de systèmes socio-techniques »

HDR présentée et soutenue à l'Académie Militaire de St Cyr Coëtquidan, le 31 janvier 2024
Unité de recherche : IRISA, UMR CNRS 6074

Rapporteurs avant soutenance :

Olivier BARAIS	Professeur, Université de Rennes
Brahim HAMID	Professeur, Université de Toulouse Jean Jaurès
Yvon KERMARREC	Professeur, IMT-Atlantique, site de Brest

Composition du Jury :

Président :	Jean-Michel BRUEL	Professeur, Université de Toulouse Jean Jaurès, IUT de Blagnac
Examinateurs :	Layth SLIMAN	Professeur, École Française de Radioélectricité, d'Électronique et d'Informatique
Garant :	Salah SADOU	Professeur, Université de Bretagne Sud, ENSIBS

Invité(s)

Marc PENNAMEN	Thales

Remerciements

Je remercie Olivier Barais, Brahim Hamid, et Yvon Kermarrec d'avoir accepté la lourde charge de rapporter cette HDR. Je remercie également Layth Sliman et Jean-Michel Bruel d'avoir bien voulu examiner ce travail, et bien sûr Salah Sadou de s'en être porté garant.

Si l'HDR est un diplôme individuel, la recherche est avant tout un travail d'équipe. Nombreux sont ceux qui, par leur aide, leur collaboration, leur confiance, leur soutien ont contribué aux travaux présentés ici.

En premier lieu, je tiens à apporter ici un témoignage de reconnaissance particulier à mes deux mentors, Eric Andonoff et JMB, qui m'ont lancé, formé et soutenu tout au long de ma carrière. Ils ont en commun un pratique assidue de la montagne, et ils ont tour à tour été de formidables premiers de cordée, des assureurs fiables sur lesquels j'ai toujours pu compter et des amis fidèles.

Toute ma gratitude va à l'Académie Militaire de St Cyr Coëtquidan qui a su me faire confiance en 2016, m'a soutenu lors de mes déboires avec ma précédente université, et m'a offert un environnement de travail riche en opportunités. Lors de cette mobilité, chance m'a été donnée d'intégrer l'IRISA, cette grande et prestigieuse fabrique à savoir, qui est un environnement particulièrement stimulant. Au sein de l'IRISA, l'ex-équipe Archware, en passe de devenir SecReizh, a été un cocon agréable et motivant, et je remercie ici tous ses membres et particulièrement Salah Sadou, Régis Fleurquin, Jamal El Hachem qui furent (et sont toujours) de précieux compagnons de travail ! Un merci encore plus particulier à Jérémy Buisson avec lequel nous avons conduits, et conduisons encore, des travaux passionnants, et cela même s'il a préféré troquer le kaki au profit du bleu.

En plus de mes activités propres à l'IRISA, mes activités de recherche à St Cyr m'ont permis de collaborer avec d'excellents collègues. Je pense ici à Lionel Touseau, Hanh-Nhi Tran, Wassila Ouerdane et Oscar Pastor pour le disciplinaire, mais également pour le trans-disciplinaire à Didier Danet, Stéphane Taillat et Saïd Haddad notamment. Qu'ils en soient ici chaleureusement remerciés.

Ils sont au départ les petites mains de la recherche, puis grandissent et deviennent autonomes avant de poursuivent sans nous. C'est un plaisir de les accompagner dans cette mue, une joie de travailler à leur coté, et une fierté de les voir voler de leurs propres ailes. Merci à mes doctorants : Natacha, Youssef, Manzoor, Imane, Nan, Paul, Angélique, Jésus et Etienne.

La recherche a besoin de carburant, qu'il soit financier, ou informationnel. Je remercie ici tous les partenaires industriels et institutionnels qui m'ont fait confiance, avec une pensée appuyée à David Hairion (Naval Group), Marc Pennamen (Thales) et au commandant Blandine (MINARM). Ces collaborations ont été possibles grâce à l'action de la Fondation St Cyr, et notamment au travail formidable du Général de Division (2S) Lafont-Rapnouil, et à l'efficacité de Thierry Renoux. Merci à eux.

Quand on regarde en arrière, on s'aperçoit que notre chemin a été guidé par nos profes-seurs. Qu'ils en soient remerciés ici. Certains ont eu une influence particulièrement forte. Je pense en particulier à Olga Bensadoum (pour son dévouement à ses élèves et ses compétences de turbodébuggeuse), à Xabi Navarro (qui m'a appris à mettre les mains dans le cambouis), et à Alain Teste (pour ses ses sujets d'examens qui me faisaient rire, et qui m'a surtout donné la chance de partir en DEA).

Comme dans toute aventure humaine, mener une carrière d'enseignant-chercheur est facilité grâce à la qualité des échanges humains que nous avons avec nos collègues plus ou moins proches. Je remercie ici particulièrement le bar des sport de Coët, notamment Laurent et Christelle pour leur soutien indéfectible, Bertrand pour sa sagesse et son rôle de punching-ball qu'il accepte de bonne grâce, Jean pour son inénarrable flot de paroles et

tous les autres ! Une pensée aussi pour mes ex-collègues palois.

Parfois, malheureusement, certains s'arrêtent en cours de route, toujours bien trop tôt. Je pense particulièrement à Vanea Chiprianov et à Séverine Sentilles. Ce fut un privilège de vous connaître et de travailler avec vous. Je pense aussi à ma mère et à mon père. Il est des absences qui sont parfois lourdes à porter.

Enfin, tout cela ne m'aurait été possible sans le soutien permanent, la compréhension et l'amour de mes proches. Nadège, qui est mon rocher, Titouan et Estéban, dont je suis si fier, je vous aime. Au final, n'est ce pas le plus important ?

Table des matières

Table des figures

Liste d'abréviations

AMSCC Académie Militaire de St-Cyr Coëtquidan

BITD Base Industrielle et Technologique de Défense

CAPEC Common Attack Pattern Enumerations and Classifications - https://capec.mitre.org/index.html

CEMA Chef d'État Major des Armées

CPE Common Platform Enumeration - https://cpe.mitre.org/

CVE Common Vulnerabilities and Exposures - https://cve.mitre.org/

CWE Common Weakness Enumeration - https://cwe.mitre.org/

ESM École Spéciale Militaire

EMIA École Militaire Interarmes

HDR Habilitation à Diriger des Recherches

HoS-ML Human oriented Security Modeling Language

IRISA Institut de Recherche en Informatique et Systèmes Aléatoires - https://www.irisa.fr/

IDM Ingénierie Dirigée par les Modèles

LII Lutte Informatique d'Influence

LIUPPA Laboratoire d'Informatique de l'Université de Pau et des Pays de l'Adour - https://liuppa.univ-pau.fr

PEC Pôle d'Excellence Cyber - https://www.pole-excellence-cyber.org/

SAS Système Auto-Adaptatif

SICS Système d'Information du Combat de SCORPION

SHS Sciences Humaines et Sociales

SoS System of Systems - Système de Systèmes

SoSTS Socio-Technical System of Systems - Système de Systèmes Socio-Technique

SysML SysML - System Modeling Language – https://www.omgsysml.org/

UBS Université de Bretagne Sud - https://www.univ-ubs.fr

UML Unified Modeling Language – http://www.omg.org/spec/UML/

UPPA Université de Pau et des Pays de l'Adour - https://www.univ-pau.fr

UT1 Université Toulouse I - Capitole - https://www.ut-capitole.fr/

UT3 Université Toulouse III - Paul Sabatier - https://www.univ-tlse3.fr/

XAI Intelligence Artificielle Explicable (Explainable Artificial Intelligence)

Chapitre 1

Introduction

Contents

Dans ce chapitre, dans un premier temps, je situe mon domaine de recherche. Dans un deuxième temps, je précise les grandes lignes de mon parcours depuis la fin de mon doctorat ainsi que ma démarche scientifique. Enfin, je donne les clés de lecture de ce document avant de présenter l'organisation du mémoire.

1.1 Du logiciel aux systèmes de systèmes socio-techniques : un cadre applicatif en évolution

En l'année 2000, exerçant alors le métier d'ingénieur d'étude, je participais au développement de la partie logicielle du "control command" d'une centrifugeuse humaine comme illustré par la Figure 1.1. Ce type de système permet d'entrainer des hommes et de tester du matériel en les soumettant à des accélérations de plus de 9 G. Ce logiciel temps-réel critique, dont nous avions la maitrise totale, fut développé en respect de la norme DO178B [Radio Technical Commission for Aeronautics, 1992]. A la fin du développement, nous livrâmes au client la documentation technique du projet, essentiellement textuelle. Elle représentait, tous les documents rédigés superposés, une pile de plus d'un mètre de haut, pour une durée de projet d'un peu plus d'un an. Outre l'obésité documentaire que cela représentait, j'ai été marqué, d'une part, par la difficulté à retrouver de l'information dans cette documentation, et, d'autre part, par le manque de support visuel sur lequel raisonner lors des phases d'ingénierie.

FIGURE 1.1 – Centrifugeuse humaine Latécoère

En ingénierie logicielle, les méthodes de développement issues de la programmation orientée objet ont formalisé, puis popularisé, l'usage de modèles sous forme de représentations diagrammatiques. Ces dernières sont en fait des artéfacts de modélisation, issus de langages reposant sur une sémantique et une syntaxe, plus ou moins précises, en support aux différentes phases de création des logiciels. Ces langages graphiques reposent principalement sur le mécanisme d'abstraction pour traiter de la complexité. On peut leur associer le vieil adage de Napoléon Bonaparte "un bon croquis vaut mieux qu'un long discours". Un effort de normalisation a conduit à la création du langage de modélisation UML dont la version 1.0 est apparue en janvier 1997 [(OMG), 1997]. Ce dernier a évolué petit à petit en fonction des retours d'expérience jusqu'à voir la création d'un incrément significatif en la version 2.0 du langage en 2004, permettant notamment une meilleure décomposition verticale des modèles.

Dans le même temps, l'idée que les modèles basés sur un langage graphique amènent une plus-value significative dans les processus d'ingénierie fait son chemin. Ainsi, une initiative de 2001 amènera l'adoption de SysML en 2005 par l'OMG. Ce dernier est dérivé d'UML2. Il est dédié à la conception des systèmes complexes. En parallèle, l'idée qu'il est possible de dériver de ces approches une ingénierie entièrement basée sur les modèles fait son chemin. Des initiatives voient le jour pour lesquelles les artéfacts principaux sont des modèles. Ces derniers s'enrichissent au fur-et-à-mesure des différentes phases du cycle de vie du produit considéré, jusqu'à parfois la création de chaines de modèles de différentes formes et préoccupations. Ces chaines de modèles conservent la sémantique des modèles amonts

via des outils de transformations de modèles. Pour les logiciels, cela peut aller jusqu'à la génération de code. Ainsi est née l'Ingénierie Dirigée par les Modèles (IDM), domaine par essence transdisciplinaire.

L'IDM aide l'ingénieur à gagner en abstraction via la réalisation de modèles avec pour objectif de mieux prendre en compte la complexification à la fois des systèmes et sous-systèmes qui constituent aujourd'hui les applications et les objets de nos quotidiens. L'approche par les modèles s'applique qu'elle que soit la taille des systèmes considérés, de la simple application sur smartphone aux véhicules divers et variés que nous utilisons sur terre, mer, air ou jusque dans l'espace. Cette complexification se retrouve dans le métier d'ingénieur pour qui, de plus en plus, les projets se retrouvent à gérer un entrelacement toujours plus grand entre les considération logicielles, matérielles, systèmes et humaines. Il convient de mettre cela en perspective dans un monde où nos systèmes au sens large sont devenus indispensables à nos sociétés, dans un contexte d'insécurité et de menaces croissantes.

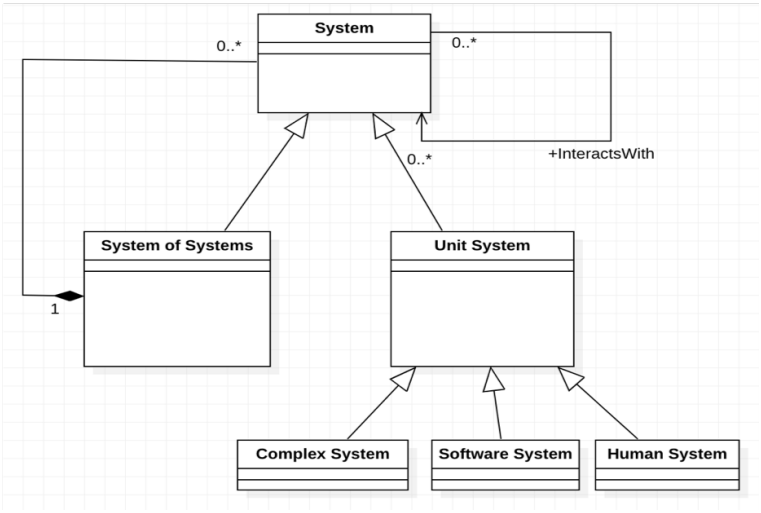


FIGURE 1.2 – Représentation de ma vision de l'imbrication entre les concepts liés à la notion de système

De cette évolution, je dresse un constat : aujourd'hui, d'une part, suite à l'évolution des systèmes et des logiciels, qui sont de plus en plus complexes et inter-connectés, et d'autre part, suite à l'évolution des outils et méthodes d'ingénierie qui prennent en compte de plus en plus d'aspects, jusqu'à intégrer les aspects humains, la séparation des cadres applicatifs (système, logiciel, humains ou données) tend à devenir obsolète, ou *a minima* moins pertinente. Ainsi, je considère le cadre dans lequel se déroulent mes travaux comme étant celui des systèmes de systèmes socio-techniques (SoSTS). Cela me permet de m'abstraire des classifications précédemment utilisées et d'évoluer dans un monde ou tout est vu comme un sous-système, comme le montre la Figure 1.2 : un système complexe est un sous-système, un logiciel est un sous-système, un opérateur humain est un sous-système. Dès lors, on peut appliquer les approches propres aux systèmes de systèmes à tous les niveaux, selon toutes les granularités. En cela, l'IDM est un excellent support puisque basée sur un haut niveau d'abstraction. Ainsi, je suis un fervent défenseur de son usage pour résoudre les problèmes d'ingénierie. Elle est donc omniprésente dans mes travaux et joue le rôle d'un fil conducteur, plus en tant qu'utilisateur convaincu qu'en tant que contributeur à l'avancée des fondements de cette approche d'ingénierie.

1.2 Parcours et démarche scientifique

Ce document s'intéresse principalement à l'évolution de mon activité de recherche depuis mon recrutement en tant que maître de conférences. La Figure 1.3 donne une vue synthétique de cette partie de ma carrière, des encadrements doctoraux que j'ai pu y mener, des domaines de recherche que j'ai traités et des types de systèmes visés. Ci-après, je commence donc par un bref rappel de ma trajectoire depuis mes débuts en recherche, puis je me concentre sur la période post-thèse.

1.2.1 La recherche n'est pas un long fleuve tranquille

Mes premiers pas dans le monde de la recherche se sont déroulés lors de mon stage de DEA effectué à l'Université Toulouse III - Paul Sabatier (UT3) en 1999 sous la direction de Gilles Zurfluh, professeur à l'UT3 et l'encadrement d'Eric Andonoff, maître de conférences à l'Université Toulouse 1 - Capitole (UT1) et de Christian Sallaberry maître de conférences à l'Université de Pau et des Pays de l'Adour (UPPA). Les travaux portaient sur la définition d'un langage d'interrogation graphique de bases de données.

J'ai ensuite exercé le métier d'ingénieur pendant presque deux ans dans une société de service toulousaine. Pendant cette période, j'ai travaillé dans le domaine naissant des applications n-tiers puis dans le temps-réel embarqué. Ces activités, même si elles relevaient de la pure ingénierie ont été fondatrices pour la suite de mes activités de recherche.

Après cette expérience industrielle, j'ai mené une thèse de doctorat au sein du Laboratoire d'Informatique de l'Université de Pau et des Pays de l'Adour (LIUPPA) sous la direction de Franck Barbier et de Jean-Michel Bruel, respectivement professeur et maître de conférences à l'UPPA. Ce travail portait sur la modélisation de la composition logicielle en UML. Nous avons proposé une définition de la composition basée sur la relation Tout ou Partie.

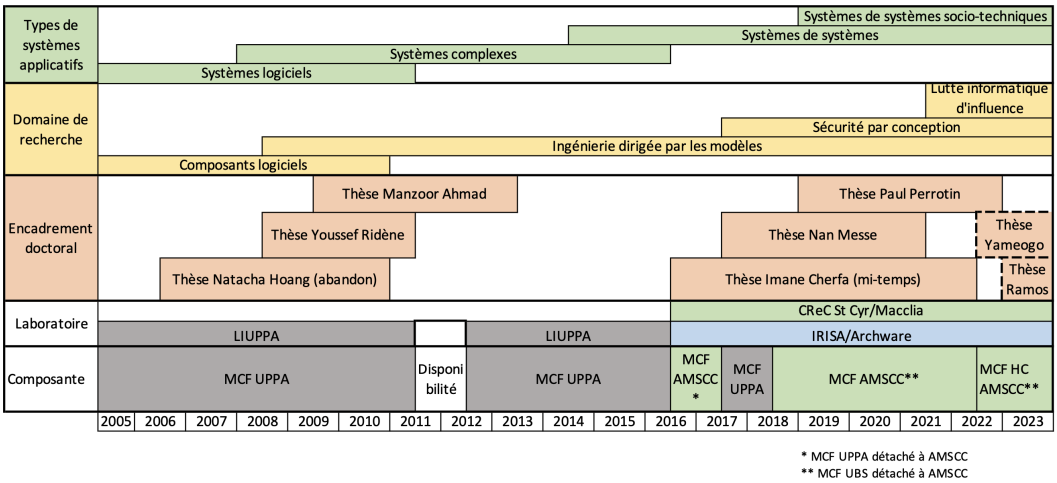


FIGURE 1.3 – Vue synthétique de ma carrière après mon recrutement en tant que maître de conférences

En 2005, j'intègre l'UPPA en tant que maître de conférences. Durant la période 2005-2010, je développe principalement deux axes de recherche en lien avec les débuts de l'Ingénierie Dirigée par les Modèles (IDM). D'une part, j'étudie les capacités du langage SysML qui vient d'être créé. D'autre part, je cherche à poursuivre mes travaux de thèse sur la

composition logicielle au niveau implémentation. Je démarre une collaboration transdisciplinaire avec Cong-Duc Pham, un professeur de l'UPPA spécialiste en réseaux. Elle porte sur la définition d'une plateforme de composition pour les réseaux de capteurs. Par la suite, en 2011, je prends une année en disponibilité pour m'occuper de mes enfants.

A mon retour en 2012, je décide de recentrer mes activités de recherche sur l'usage possible du langage SysML. Pour cela, je collabore avec Jean-Michel Bruel et l'IRIT. Nous encadrons ensemble la thèse de Manzoor Ahmad qui propose une approche pour prendre en compte les exigences pouvant être amenées à varier dans un système adaptatif. Je peux ainsi continuer à étudier l'usage des approches dirigées par les modèles à l'ingénierie système. A cette période, je travaille essentiellement avec Toulouse tout en restant à l'UPPA distante de 180 km.

Après plusieurs tentatives de mutation, en 2016, j'intègre un environnement nettement plus en adéquation avec mes activités de recherche. Je pars en détachement à l'Académie Militaire de St-Cyr Coëtquidan (AMSCC) et intègre l'équipe Archware de l'IRISA. Mais, l'UPPA refuse de renouveler mon détachement en 2017 et me rapatrie. Je réussis cependant à muter en 2018 à l'UBS et repars en détachement à l'AMSCC où je suis toujours en poste depuis cette date. Archware est une équipe spécialisée dans l'architecture des systèmes de systèmes (SoS). Mon environnement de recherche en lien avec la Défense et l'écosystème régional m'amène à m'intéresser à la sécurisation par la conception de tels systèmes. Dans ce contexte, je développe notamment avec Jérémy Buisson qui est alors maître de conférences à l'AMSCC une activité proposant d'appliquer les apports de l'IDM à certains processus militaires. J'oriente également progressivement mes travaux vers la prise en compte de la cybersécurité dans ces mêmes architectures. Enfin, les problématiques de cybersécurité m'amènent à m'intéresser à la prise en compte des aspects humains dans les modèles. Cette expertise à la croisée du monde de l'ingénierie des logiciels et des systèmes et du monde de la sécurité m'amène une certaine expertise. Elle se concrétise par le co-montage avec Jérémy Buisson de la chaire de recherche "Cyberdéfense - Cybersécurité" St-Cyr Thales dont je suis actuellement le titulaire.

1.2.2 Mode de fonctionnement

Mon activité de recherche, telle que narrée dans la section précédente, est le fruit de différentes opportunités (collaborations, financements, etc.). Cependant, plusieurs fils conducteurs peuvent se retrouver. D'une part, certainement en lien avec mon passé d'ingénieur, je me suis toujours positionné à un niveau permettant une infusion des résultats de mes travaux vers les métiers de l'ingénierie relativement rapide. D'autre part, j'ai toujours cherché à utiliser l'approche par les modèles convaincu que cela permet une meilleure compréhension des systèmes (au sens large) à réaliser. Ainsi, la majeure partie de mes travaux vise à proposer des méthodes de conception définissant des processus et s'appuyant sur des langages portés par des outils supports.

Proposer des méthodes d'ingénierie visant à “améliorer” la prise en compte de certains aspects du développement par les ingénieurs induit une difficulté, celle de montrer concrètement cette amélioration. Se pose alors le problème de la validation de l'approche proposée. En effet, montrer qu'une conception est meilleure qu'une autre n'est pas chose facile. L'approche proposée est meilleure sous quel aspect ? La facilité d'utilisation ? le nombre d'erreurs évitées ? La modularité trouvée ? La réutilisation de partie de modèles ? Il est aisé d'ajouté des propriétés à cette liste. Majoritairement, j'ai choisi de mener les validations associées à mes travaux sous la forme d'études de cas menées par des ingénieurs. Cette approche est discutable : quels ingénieurs embarquer dans nos phases de validations ? Dans le meilleur des cas, on peut faire appel à des ingénieurs expérimentés mais il est souvent difficile de

pouvoir le faire en raison des ressources contraintes auxquelles on peut faire appel dans ce cas. Une autre approche consiste à utiliser une ressource dont nous disposons de manière plus large : nos étudiants par exemple. Nous les voyons alors comme des experts juniors car souvent inscrits dans des formations à haute valeur ajoutée. L'avantage est de disposer d'un plus grand nombre d'expérimentateurs. L'inconvénient est que leur expérience est faible ce qui atténue la portée de la validation. Cet aspect de mes travaux est souvent source de difficulté, notamment pour la validation par mes pairs dans les publications scientifiques de haut niveau.

Conduire une activité de recherche requiert des moyens financiers et humains. Même si j'ai participé au montage ou à la mise en œuvre de projets, ma préférence a été jusque là majoritairement de répondre à des appels à financement de thèses pour trouver ces moyens. Je n'ai pas répondu à des appels à projets plus ambitieux tels que ceux portés par l'ANR ou l'Europe. C'est un choix assumé. L'énergie à dépenser pour le montage de tels projets, associée à la taille modeste des équipes dans lesquelles j'ai évolué, m'ont conduit à faire ce choix. Il est parfaitement pragmatique. Cela ne veut pas dire que je suis resté inactif dans la recherche de financements. Depuis 2016, j'ai été acteur principal, ou *a minima* à égalité avec mes collègues, dans la réponse aux appels à financement de thèses qui nous ont permis de prendre des étudiants en thèse de doctorat. De même, j'ai collaboré étroitement avec Jérémy Buisson dans le montage de la chaire Saint-Cyr - Thales.

Ainsi, quand je regarde en arrière, je vois mes activités de recherche comme un voyage aux multiples détours. Certains vont s'attaquer à un problème de recherche et le creuser en profondeur, à la manière des mineurs qui exploitent le nouveau filon dans la continuité de celui qu'ils viennent de terminer. Mon activité, quant à elle, a plutôt relevé du parcours en largeur, du voyage suivant le fil des opportunités qui se sont présentées.

1.3 Objectif de l'HDR

Cette habilitation à diriger des recherches (HDR) est également pour moi un moyen de faire évoluer ma pratique de la recherche vers plus de responsabilités. En effet, les besoins en encadrement sont nombreux, que cela soit à l'AMSCC, où nous n'avons pas de titulaire d'HDR en informatique, ou dans mon équipe à l'IRISA. Pour moi, le rôle d'un titulaire d'HDR est triple : (i) dans un premier temps, il doit orienter les thématiques de recherche et les thèses qui en découlent en assurant un rôle de guide, tout en assurant une mission de transmission, et cela à la fois pour le doctorant, mais également pour le co-encadrant. C'est en tout cas ce rôle de transmetteur que j'ai apprécié chez les directeurs de thèse qui m'ont fait confiance. (ii) L'autre fonction importante, toujours à mon sens, est de mettre en place les conditions financières permettant à des jeunes chercheurs de mener sereinement leurs recherches. (iii) Enfin, de par leur expérience, les titulaires d'HDR ont également le rôle d'animation et de pilotage de la recherche au sein de leurs unités. A mon sens, ces trois missions sont au centre des responsabilités que doit mener un titulaire d'HDR, et c'est là un rôle que j'entends pleinement assumer.

1.4 Organisation du mémoire

La rédaction d'un mémoire de HDR est une particularité française. Le format du mémoire d'HDR est relativement libre. Dans ce contexte, j'ai dû faire des choix que je développe ici afin de donner au lecteur quelques clés de compréhension.

L'IRISA dont je fais partie recommande de présenter une HDR relativement courte d'une longueur comprise entre 30 et 50 pages (hors articles joints et annexes) comme le

stipule la lettre de cadrage des modalités de rédaction d'une HDR, consultable en page 85. Mon style d'écriture m'a amené à déborder un peu, et je prie le lecteur de bien vouloir m'en excuser.

Dans ce contexte, je vais présenter dans ce document mes travaux en m'appuyant sur des publications significatives qui se retrouveront en annexes. Je suivrai dans la plupart des cas le canevas suivant pour chaque chapitre :

- Contexte et problématique
- Contribution
- Validation

Dans la partie "Contexte et problématique", je re-situerai le cadre dans laquelle les activités de recherche que je présente ont été menées, puis la problématique ad-hoc identifiée. Pour l'état de l'art des travaux concernés, je renverrai généralement aux articles proposés en Annexes. Dans la partie "Contribution", je ferai un résumé des contributions apportées. Enfin, dans la partie "Validation", je décrirai la manière dont les propositions portées a été évaluées.

Le mode de financement que j'ai décrit en amont a structuré d'une certaine manière mes activités de recherche. Il a été guidé également par l'évolution thématique de mon équipe qui s'est orientée vers la Cybersécurité à mon arrivée en Bretagne en 2016. On le retrouve également dans l'architecture de ce document puisque tous les chapitres des Partie I et II correspondent au travail réalisé lors d'un encadrement de thèse. Ce n'est pas le cas pour la Partie III.

Le reste du mémoire est organisé comme suit.

Dans la Partie I, je présente mes travaux portant sur les mécanismes de l'ingénierie appliqués aux systèmes et systèmes de systèmes. Plus précisément, le Chapitre 2 présente les travaux sur la meilleure prise en compte au niveau des exigences de l'adaptabilité dans les systèmes. Le Chapitre 3 synthétise la genèse du paradigme "Mission" que nous avons proposé comme étape intermédiaire de spécification des SoS.

La Partie II regroupe les travaux appliqués à la sécurité par conception. Le Chapitre 4 présente les travaux visant à réifier le concept d'"Asset" de manière à faciliter la communication entre les experts du domaine et les experts en sécurité dans la phase d'analyse de risques. Le Chapitre 5 décrit une approche visant à permettre la détection de la vulnérabilité humaine dans les architectures de SoSTS.

La Partie III quant à elle présente des travaux plus spécifiquement en lien avec le domaine de la Défense. Pour cela, le Chapitre 6 décrit des travaux exploratoires cherchant à appliquer les méthodes et outils issus de l'IDM aux méthodes et outils soutenant les opérations militaires. Le Chapitre 7 présente de son coté des travaux entrant dans le champ de la guerre informationnelle.

Enfin, la Partie IV termine ce mémoire avec un Chapitre 8 conclusion et un Chapitre 9 qui dresse des perspectives de recherche.

Première partie

Tous les chemins mènent aux modèles – Ingénierie des systèmes et des systèmes de systèmes

Les phases amont des processus d'ingénierie ont pour objectif de comprendre et de spécifier le travail à réaliser. On parle de capturer et traduire les besoins de l'utilisateur en exigences. On le sait, ces phases sont critiques et posent souvent de nombreux problèmes par la suite : mauvaise compréhension des besoins, oublis de certains besoins, exigences contradictoires, surspécifications impossibles à suivre, perte du lien avec les spécifications lors de l'évolution du système dans le temps ....

Dans cette partie, je présente deux travaux réalisés lors de deux thèses différentes. Le premier avait pour objectif de permettre l'ajout d'une certaine souplesse dans la formalisation des exigences lorsque cela était possible. Pour faire simple, comment rendre certaines exigences plus souples si elles le peuvent, et comment le formaliser dans le processus de développement ? Pour cela nous avons défini un processus basé IDM qui, à partir des exigences initiales, permet de spécifier la manière dont certaines de ces dernières peuvent être assouplies et à quelle condition, générant ainsi un ensemble de spécifications reformulées. Le second, pour sa part, se situe dans le cycle d'ingénierie légèrement en aval. Il vise à utiliser un nouveau paradigme (Mission) pour décrire de manière pérenne les exigences dans des systèmes de systèmes, en un modèle intermédiaire et antérieur à la spécification des architectures concrètes. Celui-ci a pour objet de fournir une architecture abstraite dont la réalisation concrète sera conforme aux modèles spécifiés.

Chapitre 2

Ingénierie des exigences dirigée par les modèles

Contents

Ce chapitre présente mes travaux explorant les articulations possibles entre l'ingénierie des exigences et l'ingénierie dirigée par les modèles. Dans ce cadre, nous avons mené plusieurs actions : application de l'IDM aux approches basées sur les buts, formalisation d'une approche combinée pour les systèmes auto-adaptatifs, utilisation de SysML dans l'ingénierie système dirigée par les modèles .... La thèse de de Manzoor Ahmad s'est déroulée dans ce contexte et ces travaux ont été également le cadre de collaborations nationales et internationales. L'ensemble de ces travaux a mené à la publication de plusieurs articles [Ahmad et al., 2013a, Ahmad et al., 2013b, Wanderley et al., 2014, Belloir et al., 2014, Ahmad et al., 2015].

Lors de la mise en œuvre d'un processus d'ingénierie, l'une des première étape consiste à comprendre les attentes du système à développer (les *besoins*, souvent décrits dans un cahier des charges) et à identifier les contraintes qui vont s'y appliquer, cela à la fois dans la phase de développement mais également dans la phase opérationnelle. On obtient alors une liste d'*exigences* qui servira à identifier précisément les différentes fonctions du systèmes à implémenter et ses propriétés. On parle pour cela d'*élicitation des exigences*. Ces dernières peuvent être *fonctionnelles* ou *non fonctionnelles*. Elles représentent alors respectivement ce que doit faire le système à réaliser et comment il doit le faire. L'ingénierie des exigences met l'accent sur l'utilisation de techniques systématiques et reproductibles qui garantissent l'exhaustivité, la cohérence et la pertinence des exigences du système [Sommerville and Sawyer, 1997].

Il existe un consensus arguant qu'une des raisons principales de la difficulté à correctement développer un système vient de l'incapacité des méthodes existantes à correctement identifier les exigences [Lutz, 1993]. De nombreux travaux visent à traiter ce problème dans la communauté de l'ingénierie des exigences. Dans ce contexte, nous avons développé deux axes d'effort principaux visant à explorer l'apport des techniques d'IDM à l'ingénierie des exigences. Le premier s'est intéressé à l'usage de l'IDM pour combiner plusieurs techniques de capture des exigences en lien particulièrement avec le langage SysML ; le second a focalisé sur un type de système précis, les systèmes auto-adaptatifs, afin de définir une démarche spécifique.

Les travaux décrits ici se sont déroulés dans le cadre d'une collaboration multi-niveaux. En effet, en premier lieu, alors encore au LIUPPA, j'ai développé mes activités avec Jean-Michel Bruel, professeur à l'IUT de Blagnac dans l'équipe MACAO de l'IRIT. Nous avons notamment encadré les travaux de thèse de Manzoor Ahmad. Plus largement, au niveau national, nous avons mené des actions avec l'équipe de Régine Laleau du LACL (Régine Laleau, Christophe Gnaho et Farida Semmak) et avec Raphaël Faudou, un expert industriel de ce domaine. Enfin, nous avons travaillé au niveau international avec João Araujo, professeur à l'Université Nouvelle de Lisbonne au Portugal.

2.1 Contexte et problématique

Le langage SysML est un langage de modélisation semi-formel dédié au domaine de l'ingénierie système dont la première version officielle date de 2006. Il a été développé par l'OMG en s'inspirant d'UML. Les deux langages sont actuellement aux versions 1.6 pour SysML [Object Management Group, 2019] et 2.5.1 pour UML [Object Management Group, 2017]. Outre la nature du système visé, SysML diffère d'UML en proposant des diagrammes différents, et particulièrement le diagramme des exigences qui n'existe pas en UML. Ce diagramme permet de spécifier les exigences, leurs relations, et s'articule avec un mécanisme transversal permettant la traçabilité entre les exigences et leur traduction dans les différents modèles réalisés.

Lors de sa sortie officielle, avec Jean-Michel Bruel nous nous sommes particulièrement intéressés à ce langage, et plus particulièrement au diagramme des exigences. En effet, ce dernier palliait un des manques d'UML. Il fallait dans un premier temps étudier son adéquation avec les méthodes et langages existants ou proposés en parallèle. Rapidement, les travaux de la communauté tournant autour de l'IDM ont montré tout l'intérêt de cette dernière pour tisser un lien entre méthodes et langages issus de différentes problématiques et dont le croisement des modèles résultants apportaient une plus-value notoire aux systèmes envisagés. Ainsi, il est apparu opportun de croiser l'IDM avec l'ingénierie des exigences et ses outils.

Lors de nos recherches visant à étudier l'apport de l'IDM à l'ingénierie des exigences, nous nous sommes focalisés sur un type de système particulier : il s'agit des systèmes auto-adaptatifs (SAS) dérivés des systèmes dynamiquement adaptatifs (DAS en anglais) [Whittle et al., 2009].

Choisir la nature des systèmes sur lequel les travaux de recherche vont porter n'est pas chose simple. En effet, cela peut être influencé par les partenaires avec lesquels on collabore, qu'ils soient académiques, industriels ou institutionnels. Le cadre de mes recherches, a été fluctuant comme expliqué en introduction. Les travaux menés à l'UPPA était initialement centrés sur le logiciel. Les systèmes sur lesquels j'ai appuyé mes travaux autour de SysML étaient des systèmes complexes, ou tout au moins des systèmes ayant une partie physique significative. Les travaux menés à l'UBS ou à l'AMSCC lient souvent systèmes de systèmes et systèmes socio-techniques à cause de la nature particulière des opérations militaires. Les travaux relatés ici touchent les systèmes auto-adaptables. C'est le résultat d'un choix pragmatique. En effet, à l'IUT de Blagnac, les différentes équipes de recherche collaborent autour d'un projet commun de maison intelligente[1] dédiée au maintien à domicile des personnes âgées et/ou des personnes à mobilité réduite. Ce type de maison est conçu pour être adaptable par nature. C'est donc naturellement que nous avons donc choisi d'avoir pour cible les systèmes auto-adaptatifs.

2.2 Contributions

Les contributions réalisées dans le cadre de ces recherches sont de deux sortes. La première présentée en Section 2.2.1 présente nos travaux visant à utiliser les techniques et méthodes issues de l'IDM pour tisser des liens principalement entre le langage SysML et d'autres approches de l'ingénierie des exigences. La Section 2.2.2 présente les travaux que nous avons menés en parallèle dans une thèse appliquant nos travaux aux SAS.

2.2.1 Le diagramme d'expression des exigences de SysML – Intérêt et intégration dans une démarche ad-hoc

Dans les travaux suivants, nous nous sommes intéressés au diagramme des exigences de SysML et avons étudié la manière dont ce diagramme pouvait être combiné avec différentes approches et méthodes.

Nous avons pour cela expérimenté sa nature à travers plusieurs cas d'étude, notamment en l'appliquant à la modélisation des réseaux de capteurs sans fil et sous plusieurs aspects (modélisation complète, utilisation des mécanismes d'extension ...) [Belloir et al., 2008, Bruel et al., 2009]. Par la suite, nous avons milité pour son développement via la création d'une association SysML-France, qui avait pour but d'échanger avec les industriels sur l'utilisation de ce langage et à son utilisation dans le cadre industriel. Cela nous a conduit notamment à expliquer l'utilisation du diagramme des exigences [Belloir et al., 2014].

En parallèle, nous avons approfondi l'utilisation du diagramme des exigences de SysML en lien avec des méthodes utilisées dans le cadre de l'ingénierie des exigences. Nous avons pour cela réalisé un cas d'étude [Ahmad et al., 2013a] comparant l'utilisation de SysML à des techniques telles que KAOS [van Lamsweerde and Letier, 2000], une méthode orientée but, RELAX [Whittle et al., 2009], un langage permettant d'assouplir de manière cadrée les exigences, et une combinaison de SysML et KAOS, SysML/KAOS. Cette étude nous a permis de conclure à l'intérêt de combiner une approche formalisée entre SysML et KAOS en utilisant des mécanismes d'IDM pour passer d'un modèle réalisé dans un langage à un

1. Maison intelligente de l'IUT de Blagnac https://www.iut-blagnac.fr/fr/maison-intelligente

modèle réalisé dans un autre. Nous avons de plus expérimenté l'utilisation d'une notation inspirée des “mind maps" pour saisir initialement les exigences avec les utilisateurs. En effet, [Wanderley et al., 2012] a montré que des modèles cognitifs, tels que les cartes mentales, pouvaient être utilisés en ingénierie des exigences afin de faciliter la communication entre les différentes parties prenantes. Nous avons donc proposé une méthode d'ingénierie des exigences basée IDM fonctionnant en trois temps. En premier lieu, les utilisateurs utilisent un langage de cartes mentales pour éliciter les exigences. Les modèles réalisés sont traduits en KAOS en utilisant une transformation de modèle. Cela permet aux utilisateurs de développer leurs exigences par une approche orientée but. Enfin, les modèles KAOS sont traduits en SysML par transformation de modèle de manière à intégrer les exigences exprimées dans un langage permettant par la suite de modéliser le système ainsi spécifié [Wanderley et al., 2014].

2.2.2 Une méthode d'ingénierie des exigences basée modèles pour les systèmes auto-adaptatifs

Cette sous-section résume les travaux que nous avons menés principalement dans le cadre de la thèse de Manzoor Ahmad [Ahamd, 2013] que j'ai co-encadrée avec Jean-Michel Bruel. Ils se sont déroulés de manière concomitante avec ceux rapportés dans la sous-section précédente. Ils ont ciblé comme précisé en amont les systèmes auto-adaptatifs. Ces derniers diffèrent des DAS (définis en amont) car l'adaptation du système est pilotée par le système lui-même. Cela nécessite donc de pouvoir prendre en compte la notion d'incertitude lors de la définition des exigences et, d'autre part, de disposer d'un moyen de vérifier ces exigences le plus tôt possible, avant même le début du développement de ces systèmes. Dans ce contexte, nous avons identifiées plusieurs approches intéressantes mais décorrélées. D'une part, le langage RELAX spécifié dans [Whittle et al., 2009] permet de différencier les exigences invariantes des exigences pouvant être adaptées. Il spécifie des opérateurs visant à caractériser le cadre dans lequel ces dernières peuvent varier. Cependant, ce langage n'était pas outillé. D'autre part, les langages de but tels que KAOS [van Lamsweerde, 2009] permettent une meilleure élicitation des exigences en intégrant notamment l'utilisateur dans la boucle de spécification des exigences. Une extension à SysML a également été proposée sous la forme du profile SysML/KAOS [Gnaho and Semmak, 2011] permettant d'intégrer dans le diagramme des exigences SysML les concepts de KAOS. Dans ce contexte, nous avons identifié comme objectif de proposer une méthode ad-hoc intégrant ces différentes approches en vue de mieux prendre en compte la spécification des exigences pour les systèmes auto-adaptatifs.

Le constat dressé précédemment (l'existence de plusieurs approches, non coordonnées mais pertinentes, à mettre en lien au sein d'un processus commun) nous a conduit à utiliser l'IDM pour réaliser un processus intégré. En effet, lorsqu'on dispose de deux langages supportant respectivement deux processus, avec un métamodèle spécifiant chaque langage, et que l'on peut identifier une correspondance entre les différents concepts des différents métamodèles, on peut alors définir des règles de transformation permettant de convertir tout ou partie d'un modèle exprimé selon un langage vers un autre modèle correspondant et conforme exprimé dans un autre langage.

Ainsi, nous avons défini une approche intégrée pour la modélisation et la vérification des exigences des systèmes auto-adaptables en utilisant une approche dirigée par les modèles. Pour cela, nous avons spécifié un processus qui intègre plusieurs méthodes et outils existants (SysML/Kaos, RELAX, OMEGA2/IFx). Il est décrit par la Figure 2.1. Nous avons défini des langages outillés lorsque cela était nécessaire.

Le processus spécifié commence par appliqué le processus RELAX aux exigences du

système. Cette étape fournit en sortie d'une part des exigences invariantes et d'autre part des exigences relâchées (i.e. dont l'adaptation a été formalisée à l'aide du langage RELAX). Nous avons pour cela, développé un support outillé permettant de mettre en œuvre RELAX : l'éditeur *RELAX COOL*.

Les exigences relâchées sont alors converties en but sous le profil SysML/Kaos de manière à tirer profit des approches orientées but. Pour cela, nous avons développé des règles pour transformer les exigences spécifiées par RELAX en SysML/Kaos à l'aide de règles de transformation de modèles (règles ATL) reposant sur une table de correspondance que nous avons spécifiée. A partir des modèles ainsi réalisés, après avoir étendu la définition des exigences à l'aide de KAOS, il est possible de spécifier avec SysML une première formalisation du système.

Enfin, nous avons défini un mécanisme de vérification formelle vérifiant le respect des exigences spécifiées lors du processus RELAX dans les modèles SysML réalisés avec notre approche. Celui-ci utilise le profile OMEGA2 [Ober and Dragomir, 2010] et sa boite à outils IFx. Il permet de vérifier que le modèle produit est bien conforme aux exigences initiales issues du processus RELAX.

La totalité de l'approche a été décrite dans [Ahmad et al., 2015]. Un focus particulier a été fait sur la partie validation formelle dans [Ahmad et al., 2013b].

2.3 Validation

La validation de ces travaux a été faite via la réalisation de deux cas d'études : le premier concerne la modélisation d'une maison intelligente visant à assister des personnes nécessitant une aide et une surveillance légère dans leur vie de tous les jours ; le second concerne un système de management de crise. Les deux études ont montré la faisabilité de l'approche, tant dans ses aspects de haut niveau (élicitation des exigences, utilisation des approches par but, transformations de modèles entre les différents niveaux...) que par les aspects vérification au moyen de l'approche formelle. Ils se sont avérés des preuves de concepts et ont permis de montrer que (i) l'approche était pertinente, (ii) les outils la soutenant fonctionnaient et remplissaient leur rôle, (iii) que l'approche était utilisable et compréhensible car reposant sur des concepts biens connus en ingénierie des exigences. Cependant, nous n'avons pas réussi, par manque de contacts industriels à l'époque, à appliquer notre approche à des cas industriels de plus grande envergure, ni à montrer que le retour sur investissement de ce type de méthode pouvait être intéressant.

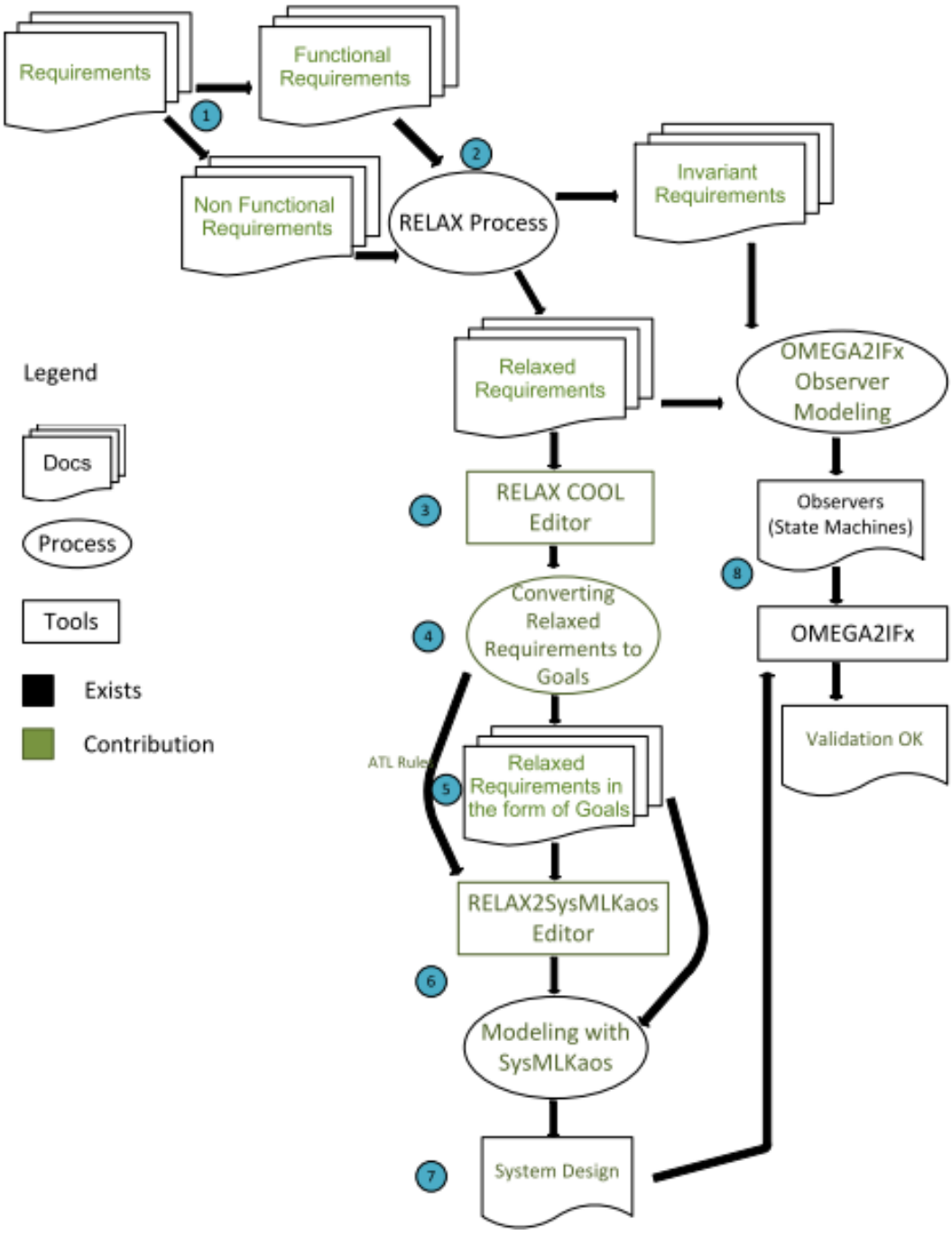


FIGURE 2.1 – Processus proposé par de la thèse de Manzoor Ahmad [Ahamd, 2013]

Chapitre 3

Ingénierie de mission basée sur les modèles pour les SoS

Contents

Les travaux décrits dans ce chapitre traitent d'un moyen d'améliorer le lien entre les experts du domaine qui interviennent en amont du processus d'analyse et les architectes systèmes qui mettent en œuvres ces décisions lors de l'analyse et la conception du système. Dans le contexte des systèmes de systèmes, plus que tout autre, le système réalisé va évolué au cours de sa vie et peut voir sa forme s'éloigner de manière plus ou moins maitrisée de sa spécification initiale. Aussi, nous avons proposé l'utilisation d'un paradigme nouveau, le concept d'ingénierie de mission, pour préserver le lien entre les spécifications initiales et la mise en œuvre architecturale tout au long de la vie du système de système. Pour cela, nous avons proposé une approche orientée modèle permettant d'expliciter dans les modèles l'ingénierie de mission. Ces travaux ont été menés dans la cadre de la thèse d'Imane Cherfa [Cherfa, 2022] et ont conduit à la publication de plusieurs articles [Cherfa et al., 2018, Cherfa et al., 2019].

Après avoir exploré l'utilisation des modèles dans le cadre de l'ingénierie des exigences, les travaux présentés ici explorent l'articulation entre la spécification issue des exigences et la conception, plus spécifiquement au niveau de l'architecture. Le domaine d'application que nous traitons est celui des systèmes de systèmes (SoS). Dans ce type de système, la durée de vie peut être longue amenant les architectures physiques à évoluer parfois en perdant le lien avec la spécification initiale.

3.1 Contexte et problématique

Les SoS sont des ensembles de systèmes autonomes et indépendants qui interagissent et coopèrent pour atteindre des objectifs communs. Ces entités peuvent être de natures diverses (systèmes complexes, systèmes humains ...). Une de leurs particularités est que chaque sous-système (on parlera également de système constitutif) peut évoluer unilatéralement au cours de la vie du SoS, risquant ainsi de ne plus correspondre à la fonction que celui-ci devait fournir au système complet. Une des raisons à cela vient du fait que les SoS, de par leur nature, peuvent avoir une durée de vie longue à très longue, avec le risque notamment que les architectes systèmes pilotant les évolutions de chaque sous-système s'éloignent des spécifications initiales du SoS. Nous nous situons ici dans une sous-famille des SoS, dans laquelle on considère pouvoir bénéficier d'un processus d'ingénierie descendant dans lequel il est possible de spécifier un certain nombre de préoccupations au niveau du SoS, ces spécifications impactant les sous-systèmes.

Conformément à cette vision, nous avons déterminé le besoin de disposer d'une spécification de haut niveau jouant le rôle d'un invariant attendu du système. Cette spécification a pour rôle de fournir des modèles suffisamment abstraits pour que l'architecte système puisse librement choisir les systèmes constituants à utiliser, et suffisamment explicites pour contenir les attendus du SoS. En d'autres termes, il est nécessaire de disposer d'une modélisation intermédiaire entre les exigences du SoS et l'architecture abstraite du SoS, intégrant ainsi les invariants attendus de ce dernier.

L'approche retenue a été d'utiliser l'ingénierie de mission [Sousa-Poza, 2015], tirant son inspiration initialement des opérations militaires et dont la philosophie est de spécifier ce que le système doit faire, la mission qu'il doit remplir, que l'on considère comme un invariant. L'ingénierie de mission vise à établir un lien entre les activités d'ingénierie menées pour réaliser une mission et la mission elle-même.

Il existe dans la littérature plusieurs approches proposant des outillages et des méthodes diverses visant à intégrer les principes de l'ingénierie de mission dans une forme plus large d'ingénierie, et plus particulièrement dans l'ingénierie des SoS. Nous proposons pour notre part d'intégrer l'ingénierie de mission dans un processus d'ingénierie système dirigé par les modèles.

3.2 Contributions

Nos travaux sur le sujet ont conduit à la définition d'un processus orienté mission basé sur les modèles, intégrable à un processus plus large orienté SoS. Il se nomme *MOP-SoSE* (Mission Oriented Process for System of Systems Engineering). Ces contributions ont été principalement produites dans la thèse d'Imane Cherfa [Cherfa, 2022] et sont résumées dans [Cherfa et al., 2018, Cherfa et al., 2019]. Plus particulièrement, elles prennent la forme d'un modèle conceptuel décrivant les éléments caractéristiques d'une mission. En parallèle, nous avons défini le processus MOP-SoSE.

Les concepts clés sur lesquels repose le processus sont les suivantes : (i) Le processus s'applique aux SoS dans lesquels une entité gère et pilote la mise en œuvre du SoS. Des entités et des équipes d'ingénierie systèmes indépendantes sont responsables des systèmes constitutifs. (ii) Un SoS est considéré comme un environnement dans lequel les systèmes constitutifs évoluent, pour accomplir une mission donnée. L'environnement du SoS est donc incertain, car au cours d'une mission, et en fonction du contexte de cette dernière, l'attribution des fonctions aux sous-systèmes peut changer : différents systèmes constituants peuvent alors être ajoutées, supprimées ou modifiées dans/depuis le SoS. (iii) Le contexte de la mission détermine le fil de la mission, puis le contexte de la mission aide à déterminer les fonctionnalités nécessaires et les systèmes constituants à impliquer. (iv) La mission de bout en bout est considérée dans notre processus comme la mise en œuvre d'un système, dans laquelle l'architecture est générée de manière aussi automatique que possible, afin d'éviter la perte d'informations entre l'expert du domaine et l'architecte du système. L'architecture concrète est élaborée à partir d'une architecture abstraite. Cette dernière sert d'invariant qui guide le choix des architectes systèmes en charge de la mise en place des systèmes physique au sein du SoS.

3.2.1 Un modèle conceptuel de mission

Dans un premier temps, nous avons formalisé le concept de mission dans le cadre d'un modèle conceptuel. Cela a pour objectif d'identifier et de définir les notions clés et les liens entre elles. Nous avons, en particulier, défini les concepts suivants : (i) les exigences de la mission qui définissent la mission sous forme de sous-missions composites et de missions atomiques ; (ii) la vue opérationnelle qui décompose les missions atomiques en activités paramétrables : (iii) les systèmes constituants qui jouent le rôle des actions décrites par les activités. La première version de ce modèle conceptuel a été décrite dans [Cherfa et al., 2019] puis affinée dans [Cherfa, 2022].

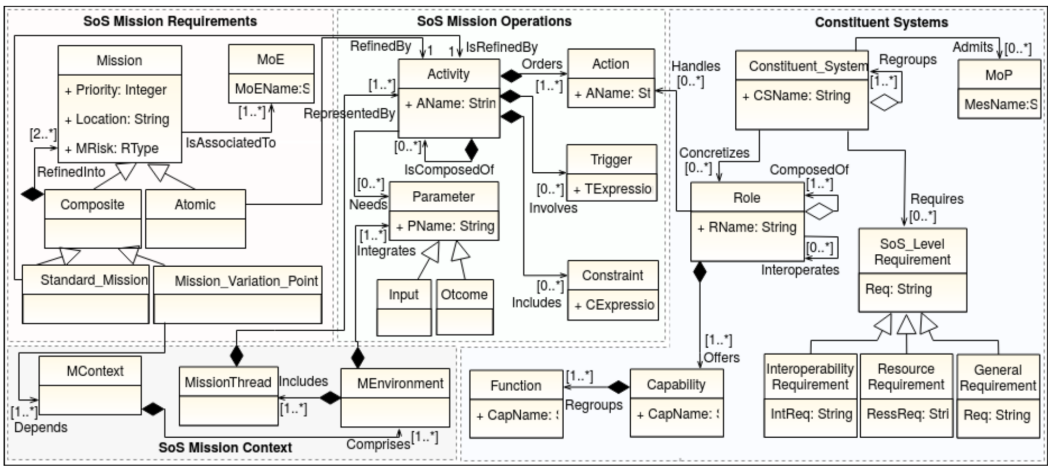


FIGURE 3.1 – Modèle conceptuel de mission [Cherfa, 2022]

3.2.2 MOP-SoSE : un processus orienté mission pour les SoS

Le cycle de vie porté par le processus MOP-SoSE est une adaptation du modèle en vague pour les SoS (SoSE wave model). En effet, ce modèle nous a semblé le plus pertinent

puisqu'il permet la prise en compte continue d'adaptations et ainsi une évolution maitrisée de l'architecture. Dans ce contexte, nous avons identifié plusieurs acteurs jouant un rôle différent dans le processus. Au niveau du SoS, on peut noter le responsable de la mission, l'expert applicatif du domaine, et l'architecte système. Côté système constituant, on s'intéresse principalement aux ingénieurs systèmes. Enfin, d'un point de vue mise en œuvre, nous avons choisi d'utiliser SysML comme langage de spécification et de modélisation.

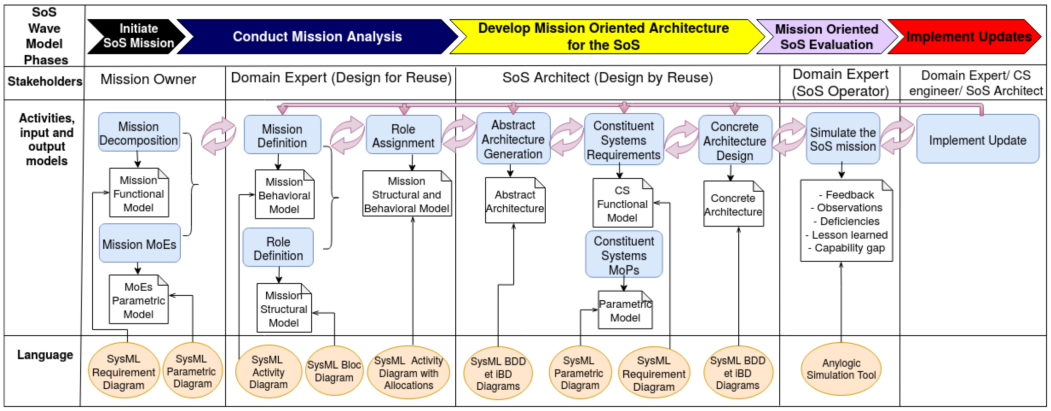


FIGURE 3.2 – Processus MOP-SoSE [Cherfa, 2022]

Dans ce contexte, le principe retenu consiste à spécifier la mission du SoS à l'aide de modèles structuraux et comportementaux. En raffinant le modèle de mission, un modèle d'architecture abstrait est généré à l'aide de transformations de modèles. Le modèle obtenu intègre les spécifications de la mission. A partir de ce modèle d'architecture abstraite, une architecture concrète est mise en place par les architectes systèmes des systèmes constituants, en conformité avec l'architecture concrète. La Figure 3.2 illustre ce processus. Plus précisément, nous proposons les étapes suivantes.

En premier lieu, le spécialiste de la mission va décomposer cette dernière de manière à fournir une vue fonctionnelle à gros grain de la mission. Cet aspect est obtenu par une analyse des objectifs généraux de la mission afin d'identifier de manière récursive des objectifs secondaires plus précis. Le critère d'arrêt de la décomposition de la mission est l'identification d'un processus capable d'exécuter une sous-mission donnée.

Ensuite, il va définir les mesures d'efficacité de la mission. Ces mesures seront utilisées pour évaluer la performance globale de la mission réalisée.

L'expert du domaine va ensuite conduire l'analyse de la mission. Pour cela, il va dans un premier temps la définir en tant que vue opérationnelle en identifiant les éléments et les activités de la mission de manière à aboutir à une vue comportementale fine des sous-missions.

Dans un deuxième temps, il va identifier les rôles de la mission. Ces derniers sont utilisés pour fournir une représentation abstraite de la hiérarchie des entités ayant des capacités qui permettent d'accomplir la mission. Les capacités peuvent être fournies ou exigées par les rôles, ce qui permet la composition des rôles.

Dans un dernier temps, il affecte les rôles à des activités qui sont composées d'actions qui correspondent aux capacités des rôles. Le rôle est composé de plusieurs capacités, et la même capacité peut apparaître dans différents rôles. Par conséquent, cette étape vise à désigner le rôle qui doit être associé à chaque action d'une activité.

Par la suite les ingénieurs systèmes prennent la main. Dans un premier temps, l'ingénieur

système du SoS va générer un modèle d'architecture abstraite de la mission. L'architecture est une vue structurelle qui décrit les systèmes constitutifs du SoS et leurs connexions. Cependant, toutes les définitions susmentionnées ne se réfèrent qu'aux rôles et non aux systèmes constitutifs. Par conséquent, la première architecture générée à partir des définitions données correspond à l'architecture abstraite du SoS.

Avant de remplacer les rôles par des systèmes constitutifs concrets, l'architecte peut identifier de nouvelles exigences nécessaires sur ces derniers à leur intégration.

Des mesures de performance (MdP) sont décrites pour chaque service de l'architecture, afin de déterminer les capacités et les limites de tous les systèmes constituants concernés. Cela permet de choisir le meilleur système critique pour traiter une action donnée.

L'architecture abstraite est progressivement affinée pour obtenir l'architecture concrète par les ingénieurs systèmes. Une simulation peut être jouée pour évaluer la capacité d'une configuration d'architecture à accomplir la mission SoS spécifiée. Elle permet également de confirmer les performances et de découvrir les erreurs.

Le SoS est mis en œuvre. Les mises à jour peuvent être effectuées au niveau des modèles du SoS ou au niveau des systèmes critiques.

La formalisation du processus MOP-SoSE a nécessité la définition d'un modèle conceptuel de mission. Nous avons implémenté les concepts de ce modèle à l'aide d'un profil SysML. Ce choix résulte de notre volonté de, d'une part, maximiser dans le processus l'utilisation de diagrammes et de concepts natifs à SysML, d'autre part de la volonté de rester au plus prêt d'un langage largement utilisé dans l'industrie.

3.3 Validation

Afin à la fois d'illustrer notre approche et de la vérifier, nous avons mené de bout en bout un cas d'étude. Celui-ci consistait en la spécification et la modélisation d'un système de gestion de foule [Gorod et al., 2014]. Pour cette étude de cas, après avoir identifié une des missions à prendre en compte pour une tel SoS, nous avons mené une approche itérative de spécification de la sous-mission "*surveillance de la foule dans les stades*" : vue fonctionnelle, vue structurelle, performances requises. Nous avons par la suite généré son architecture abstraite, et donné un exemple d'une architecture concrète correspondante.

Nous avons pu constater plusieurs points positifs. Dans un premier temps, la planification et la préparation, permettent une bonne connaissance de la mission. Ensuite, à travers la définition des rôles, il résulte une bonne connaissance des sous-systèmes qui seront amenés à intervenir à un moment donné lors de l'évènement. On peut ainsi anticiper certaines coopérations entre des rôles qui ne sont pas destinés, à la base, à travailler ensemble, afin de maximiser l'efficacité de la mission. Enfin, l'approche imposant des améliorations constantes du SoS, les solutions technologiques et logistiques améliorant l'efficacité de la mission doivent être recherchés et adoptées de manière continue, ce que le processus permet.

Pour pouvoir évaluer les différents modèles, nous avons mené une simulation. Elle repose sur trois scénarios principaux : "*surveillance de l'entrée de la foule*", "*surveillance du comportement de la foule après l'entrée*" et "*surveillance de la foule lorsque les piétons quittent le stade*". Nous avons noté une amélioration continue des mesures d'efficacité de la mission à chaque itération, jusqu'à stabilisation. Après chaque itération, des révisions ont été faites dans les modèles afin d'améliorer la prise en compte de la mission.

Cette étude à été menée à l'Université de Blida 1, avec deux étudiants de dernière année et la doctorante, cela sur une durée de 6 mois chacun.

Deuxième partie

A la croisée des chemins – Ingénierie de la sécurité par conception

Cette partie décrit les travaux marquant mon inflexion thématique vers le domaine de la cybersécurité. Ce dernier est généralement traité à travers deux points de vue : *a priori* ou *a posteriori*. L'approche *a posteriori* est la plus développée. On peut la résumer de manière très succincte en un déploiement d'une défense périmétrique centrée sur la protection des réseaux et des systèmes à travers la mise en place d'une infrastructure sécurisée. Pour cela, on s'appuie sur des éléments techniques tels que le déploiement de pares-feux, de mécanismes de contrôle d'accès, de procédures et de contrôles tels que la surveillance d'intrusions, ...

La stratégie *a priori*, telle que la "sécurité par la conception", a historiquement été moins considérée [van den Berghe et al., 2018]. Du point de vue de l'ingénierie logicielle et système, la sécurité est historiquement une propriété non-fonctionnelle parmi d'autres. Force est de constater qu'aujourd'hui, pour de nombreuses raisons (surface d'exposition de plus en plus grande, nombre de cyberattaques croissant exponentiellement, ...) la prise en compte de cette propriété doit fortement être accrue dans le développement des méthodes d'ingénierie.

Ce constat fait, mon intérêt pour cette thématique a dans un premier temps été guidé par opportunisme. En effet, la région Bretagne est un des fers de lance de la cybersécurité en France. A ce titre, l'écosystème régional soutient avec force son développement. Dans ce contexte, le Pôle d'Excellence Cyber (PEC), qui a financé une partie de mes travaux, mais aussi l'UBS qui a fait de la cybersécurité un de ces axes prioritaires, ont été des acteurs majeurs m'incitant à prendre cette direction.

L'inflexion thématique décrite ici n'a cependant pas été une révolution sur mon cœur de recherche. En effet, j'ai intégré cette problématique et l'ai traitée en utilisant les mêmes approches que celles que j'utilisais précédemment, à savoir en définissant des méthodes proposant des processus d'ingénierie et des langages basés modèles les supportant. Par contre, ce qui a été nouveau, et qui transparait dans la manière dont j'ai traité cette problématique, c'est que cela a accru mon intérêt à supporter la prise en compte de l'humain dans le processus d'ingénierie. En effet, dans le chapitre 4, nous nous sommes focalisés sur la manière dont on pouvait faciliter les échanges entre spécialistes issus de différentes communautés (ici les experts d'un domaine et les experts en sécurité). Dans le chapitre 5, le sujet traité est directement la représentation de l'humain dans un processus d'ingénierie. Ce travail est probablement l'origine séminale de mon intérêt pour une meilleure prise en compte du traitement de l'humain dans la modélisation. On retrouvera cette marque également en partie 3 de ce document.

Chapitre 4

Rapprochement des architectes et des experts sécurité

Contents

Ce chapitre présente nos travaux visant à améliorer la coopération entre experts du domaine et experts en sécurité lors du processus de conception des architectures systèmes. Pour cela, nous avons proposé de réifier le concept “d’asset” à travers un métamodèle. Ce métamodèle sert ensuite de pierre angulaire à un framework d’assistance à la prise en compte de la sécurité dans la construction des architectures. Dans un deuxième temps, ce métamodèle est utilisé comme support à la structuration de la phase d’analyse des menaces. Ce travail a conduit à la publication de trois articles [Messe et al., 2019, Messe et al., 2020b, Messe et al., 2020a]

Prendre en compte la sécurité lors du développement d'un projet n'est pas chose aisée. Même si on sait que plus une exigence est formulée tôt dans le processus de développement, meilleure sera sa prise en compte. Savoir exprimer clairement le niveau de sécurité souhaité est délicat lors de la phase d'élicitation des exigences. La sécurité relève à notre avis davantage d'un point de vue micro que d'un point de vue macro. En effet, il faut déjà avoir une vue assez claire de la manière dont le système sera constitué pour définir efficacement les mécanismes de sécurité à associer au développement de ce dernier. C'est pourquoi nous pensons qu'une attention particulière sur la sécurité doit être apporté lors de la phase de conception architecturale. C'est là que nous situons les travaux présentés dans ce chapitre.

4.1 Contexte et problématique

La sécurisation par la conception implique de mettre en relation deux compétences. La première est classique, il s'agit de la conception architecturale qui dépend d'un "architecte". Celui-ci a pour rôle de concevoir l'architecture du système à réaliser. Pour cela, il dispose d'une connaissance à la fois du métier et de la manière de construire un système. Généralement, on considère que les architectes ont peu de connaissances en sécurité et doivent donc faire appel et interagir avec des experts en sécurité. La compétence de ces derniers consiste à savoir déployer des procédures et des moyens techniques assurant la sécurité au sein des architectures. Pour cela, ils pensent l'architecture à travers une vision attaquant.

Cette mise en relation n'est pas toujours faisable car elle peut prendre du temps (et donc cela peut allonger la durée nécessaire à la mise sur le marché par exemple), être couteuse (et rentrer en contradiction avec par exemple des exigences de performances et/ou de concurrence), être complexe pour des entreprises de petite taille ...

D'autre part, afin de rendre cette mise en relation efficiente, il faut être conscient que la connaissance nécessaire à la prise en compte de la sécurité dans les architectures évolue quotidiennement. Il faut donc que, lors de cette mise en relation entre l'architecte et l'expert en sécurité, ces derniers puissent s'appuyer sur ces connaissances actualisées. De grandes bases de connaissances recensent ce savoir. On peut par exemple citer celles fournies par Mitre corporation [1] qui font partie des références sur la question :

CAPEC Common Attack Pattern Enumerations and Classifications : recense les patrons d'attaque connus. Les patrons d'attaque sont des descriptions générales de scénarios d'attaque qui sont souvent utilisées pour exploiter les vulnérabilités dans un système.

CPE Common Platform Enumeration : fournit une description commune et formalisée des systèmes et sous-systèmes constituants informatiques via un système de dénomination structuré. Cela permet une sorte de "typage" des éléments architecturaux notamment.

CVE Common Vulnerabilities and Exposures : recense les vulnérabilités connues. Une vulnérabilité est une faille dans un système ou une application qui peut être exploitée par un attaquant pour accéder à des informations confidentielles, endommager le système ou prendre le contrôle du système. Les vulnérabilités sont souvent le résultat d'erreurs de programmation, de configurations incorrectes ou de bogues dans le logiciel.

CWE Common Weakness Enumeration : recense les faiblesses de sécurité connues. Une faiblesse est un problème de conception ou de mise en œuvre qui peut rendre un système ou une application plus vulnérable aux attaques. Les faiblesses peuvent

1. Mitre corporation : Organisation à but non lucratif américaine https://www.mitre.org/.

inclure des pratiques de codage faibles, une configuration de sécurité inadéquate, des erreurs de conception ou des erreurs de configuration.

On le voit, intégrer la sécurité lors de la conception des architectures impose de faire croiser deux types d'expertise assez différentes et une connaissance à jour de la sécurité, et plus particulièrement des faiblesses, des vulnérabilités et des attaques connues. Or, force est de constater que les méthodes existantes ne le supportent que faiblement d'une part, et que d'autre part la différence de culture entre les deux familles d'acteurs est un frein à une bonne communication.

Dans ce contexte, nous avons identifié le besoin de spécifier une méthode intégrant les intérêts des deux familles d'expertise (la description d'une architecture soutenant un logiciel et/ou un système d'une part avec la recherche et l'identification de moyens de l'attaquer d'autre part) afin de les assister dans leur collaboration de manière semi-automatisée.

Ces travaux se sont déroulés dans le cadre du Pôle d'Excellence Cyber [2], qui a financé la thèse de Nan Messe [Messe, 2021] sur le sujet, en collaboration avec la Direction Générale de l'Armement [3] et sa branche Maîtrise de l'Information.

4.2 Contributions

Nos travaux sur ce sujet ont conduit à deux contributions décrites en aval. Nous avons d'abord proposé de réifier le concept de *bien* (*asset*), puis l'avons intégré dans une framework orienté sur la coopération entre les deux familles d'acteurs. Dans un second temps, nous avons proposé de structurer le processus d'analyse des menaces en s'appuyant sur le framework proposé.

4.2.1 Un framework basé “asset" pour l'amélioration de la coopération entre architectes et experts en sécurité

Dans un premier temps, nous avons modélisé l'univers que représente d'une part les concepts manipulés par les architectes et d'autre part ceux permettant leur sécurisation, de manière à identifier les points de convergence. Pour cela, nous avons réifié le concept de “bien”, le considérant comme un creuset commun aux différents intervenants dans le processus. Nous considérons en fait trois types de biens selon le point de vue adopté. L'expert du domaine est capable de définir les biens ayant le plus d'importance dans son architecture. Nous les appelons les *biens du domaine*. De son coté, en identifiant les faiblesses de l'architecture, l'expert en sécurité peut spécifier ce qui a de la valeur pour lui, c'est-à-dire les biens qu'il peut attaquer. Nous les appelons les *biens vulnérables*. En fusionnant les deux, on identifie les biens métiers qui présentent une vulnérabilité, c'est-à-dire un intérêt pour l'attaquant. Nous les appelons les *biens du domaine vulnérables*. Ce sont ces derniers qu'il faudra protéger en priorité. Ainsi, en utilisant la notion de *bien* comme pivot, nous avons proposé un moyen d'identification des vulnérabilités à traiter dans l'architecture. Cette vision est schématisée dans la Figure 4.1.

En nous basant sur ce principe, nous avons proposé un framework d'assistance à la prise en compte de la sécurité au sein d'une architecture. Pour cela, nous avons d'abord défini un modèle de données commun mettant en relation les différents concepts nécessaires [Messe et al., 2020b]. Il est structuré autour de trois points de vues : la vue du domaine, la vue attaquant et la vue défenseur. Il a la particularité de pouvoir traiter à la fois des architectures abstraites, mais également des architectures plus concrètes. En effet, certaines

2. Pôle d'Excellence Cyber : https://www.pole-excellence-cyber.org/
3. DGA : https://www.defense.gouv.fr/dga

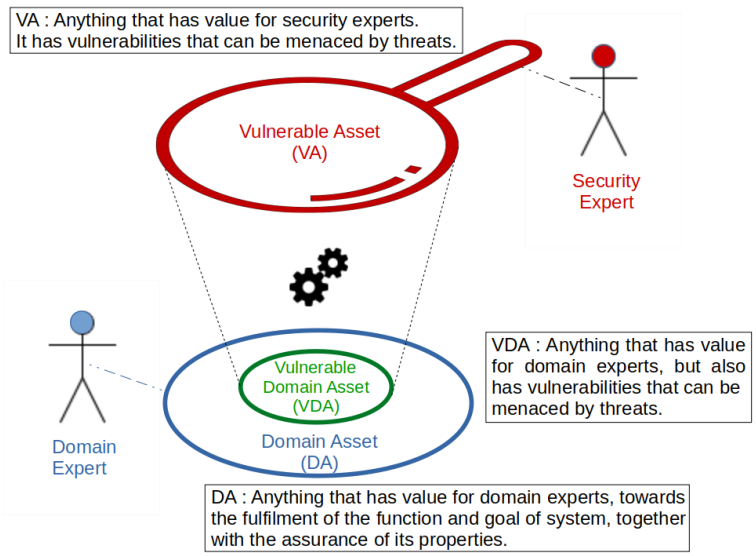


FIGURE 4.1 – Approche de réification du concept de "bien" [Messe, 2021]

décisions architecturales peuvent être prises en compte à un haut niveau d'abstraction (mise en place de patrons de sécurité par exemple) quand d'autres doivent être prises avec une vue concrète de l'architecture déployée (application de tel patch correctif sur telle version d'un serveur par exemple). Pour cela, nous avons défini un processus itératif d'assistance. Il est structuré en deux phases. Dans la première, l'expert du domaine annote ses modèles d'architecture en identifiant les biens métiers, les propriétés de sécurité qu'il souhaite leur appliquer (les propriétés standards : disponibilité, intégrité, confidentialité), et la catégorie sur laquelle cela s'applique (donnée, comportement du système, lien avec l'humain et bien physique). Pour chaque bien annoté, l'expert en sécurité va essayer de trouver un bien vulnérable correspondant. S'il y arrive, il devient alors un bien métier vulnérable et les deux acteurs recherchent alors une contremesure applicable. Le processus est illustré par la figure 4.2.

Le processus d'assistance s'appuie sur les bases de connaissances telles que celles proposées par Mitre Corporation. Cependant, nous avons été confronté à un problème. En effet, la plupart sont constituées de fiches documentaires plus ou moins rédigées en langage naturel. Il existe bien une structuration entre les bases ainsi que des éléments de traçabilité entre les fiches, mais leur exploitation automatique est compliquée à mettre en œuvre. Aussi, nous avons dérivé un modèle de données à partir du modèle conceptuel et nous avons extrait le contenu des fiches issues des bases documentaires afin de les intégrer à une base de données ainsi constituée et plus aisément exploitable. Le peuplement de la base de données est grandement manuel. Ce n'est pas optimal mais nous assumons le fait que notre approche le justifie puisqu'elle montre qu'en structurant l'information à l'image de notre modèle, nous améliorons l'exploitation des fiches issues des bases de connaissance et leur traitement.

4.2.2 Une proposition de structuration du processus d'analyse des menaces

Dans un second temps, nous nous sommes intéressés à l'intégration du framework d'assistance proposé dans un processus de modélisation des menaces industriel, tel que Microsoft

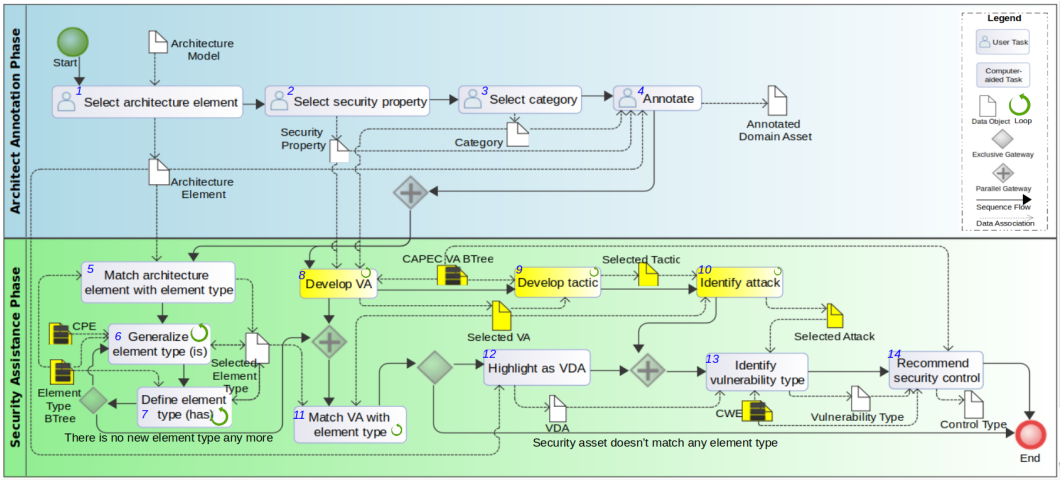


FIGURE 4.2 – Processus d'asssistance à la sécurisation des architectures [Messe et al., 2020b]

SDL. Ce dernier est basé sur STRIDE, qui est actuellement la méthode de modélisation des menaces la plus aboutie [Shevchenko et al., 2018], et est mis en œuvre avec l'outil de modélisation des menaces SDL, qui est disponible en ligne [Microsoft Corporation, 2018]. Il comporte quatre étapes, décrites dans [Shostack, 2008] : 1) la schématisation, 2) l'énumération des menaces (en appliquant STRIDE), 3) l'atténuation et 4) la vérification. Comme notre approche vise à soutenir l'énumération des menaces, nous nous sommes limités aux deux premières étapes : la première consiste à modéliser le système à l'aide de Data Flow Diagrams (DFD) ; la deuxième, principalement basée sur du "brainstorming" consiste à identifier les menaces.

Notre apport se situe notamment à deux niveaux. (i) Premièrement, l'approche par le DFD est centrée sur les données. Un DFD se concentre sur le flux de données entre les composants du même niveau d'abstraction. Chaque nouveau modèle, qui peut être créé avec l'outil, peut représenter un nouveau niveau d'abstraction. Cependant, le DFD ne permet pas de présenter les relations entre les éléments de différents niveaux d'abstraction. C'est pourquoi nous avons proposé de modéliser le système à l'aide d'un diagramme de classes UML, qui permet de modéliser des éléments de différents niveaux d'abstraction. (ii) L'autre apport de notre approche est de structurer la phase 2 de SDL qui consiste en un brainstorming non structuré. Le résultat de ce brainstorming est fortement corrélé au niveau d'expertise des personnes impliquées dedans. En outre, la non structuration montre, qu'en pratique, certaines parties du systèmes ne sont pas prises en compte. La structuration que nous avons proposée permet une systématisation du processus à tous les actifs, et cela y compris en prenant en compte plusieurs niveaux d'abstraction. L'étude de cas que nous avons menée a montré que nous identifions plus de vulnérabilité qu'avec SDL seul.

4.3 Validation

Afin de valider notre approche, nous avons, dans un premier temps, mené une étude de cas *ad-hoc* avec notre approche, qui nous a permis d'identifier 14 menaces précises là où l'utilisation de SDL permettait l'identification de seulement 2 catégories de menaces [Messe et al., 2020a]. De plus, le fait que notre approche puisse prendre en compte plu-

sieurs niveaux d'abstraction dans l'architecture est une vraie valeur ajoutée par rapport à l'approche SDL qui impose un même niveau conceptuel lors de sa première étape.

Dans un deuxième temps, nous avons mené une expérimentation [Messe, 2021] basée sur un processus de conduite expérimentale nommé "Crossover", très répandu dans les expérimentations en génie logiciel [Vegas et al., 2015]. L'objectif était d'utiliser l'approche avec deux groupes témoins différents, cela pour notamment éliminer certains biais. En effet, cette méthode permet de prendre en compte le risque de niveau de connaissances différent entre les acteurs de l'expérience antérieure à celle-ci. Cette approche permet de contrôler la variabilité entre les sujets.

L'expérimentation a montré que, du point de vue de la qualité, notre outil d'assistance à la sécurité identifie les menaces pertinentes aussi bien que l'outil SDL de Microsoft. Il présente toutefois l'avantage d'afficher les informations sur les vulnérabilités et les contrôles de sécurité, alors que l'outil SDL de Microsoft ne le fait pas. D'un point de vue qualitatif, notre assistance à la sécurité fournit plus de types d'informations (vulnérabilité et contrôle de sécurité) que l'outil SDL de Microsoft. Les résultats quantitatifs montrent que l'assistance de sécurité identifie en général plus de menaces que l'outil SDL de Microsoft, car la référence des données de l'assistance est basée sur CAPEC et CWE, qui contiennent plus de détails que STRIDE, la base de données de référence utilisée par l'outil SDL de Microsoft.

Enfin, nous avons soumis le panel d'utilisateurs impliqués dans l'expérimentation à un questionnaire portant sur l'utilisatbilité de notre approche. Il en ressort que (i) l'approche est jugée facile à utiliser à plus de 57 %, (ii) les instructions pour la mettre en œuvre sont faciles à comprendre à plus de 70 %, (ii) la facilité à exploiter les résultats bonne à plus de 57 % et enfin (iv) la facilité à comprendre ces mêmes résultats est très bonne à 100 %.

Chapitre 5

Détecter la vulnérabilité humaine dans les SoSTS

Contents

Les travaux présentés dans ce chapitre résument une contribution visant à permettre la détection de la vulnérabilité humaine dans les modèles d'architecture de systèmes de systèmes socio-techniques. Ces travaux ont été menés dans le cadre de la thèse de Paul Perrotin [Perrotin, 2022] et ont été publiés dans [Perrotin et al., 2022a, Perrotin et al., 2022b]. Ils rentrent également dans le contexte d'une collaboration avec la Chaire de Cyberdéfense des Systèmes Navals [a] et Naval Group [b].

a. Chaire de Cyberdéfense des Systèmes Navals : `https://chaire-cyber-navale.fr/`

b. Naval Group : `https://www.naval-group.com/fr`

Dans la majorité des attaques cyber qui réussissent, les vulnérabilités qui sont initialement exploitées ne sont pas issues d'un système technique, mais bien de l'action d'un opérateur humain. Ce dernier, par négligence ou par une méconnaissance des règles de bonne pratique par exemple, va commettre une erreur permettant à une attaque de réussir [Verizon, 2016]. Il y a donc un enjeu considérable à réduire cette surface d'attaque. Anticiper une vulnérabilité humaine est donc une capacité importante dans l'arsenal des experts en sécurité. Or, s'il existe de nombreux travaux portant sur la détection de vulnérabilités techniques, ce n'est pas le cas pour la détection de vulnérabilité humaine.

Dans ce chapitre, fidèles à l'approche que nous développons dans nos recherches, nous proposons une méthode d'ingénierie en ce sens. L'idée défendue est qu'il est possible de diminuer par construction le nombre de vulnérabilités humaines possibles dans un système ou dans un SoS. Pour cela, notre approche propose une méthode visant à déterminer si les opérateurs humains d'un système ou d'un SoS présentent une vulnérabilité qui pourrait mener à une défaillance lors d'une attaque cyber, ceci conceptuellement. Pour cela, il faut dans un premier temps être capable de modéliser le ou les opérateur(s) humain(s) dans le système étudié, puis de développer un moyen d'identifier la présence potentielle d'une vulnérabilité dans le système modélisé.

5.1 Contexte et problématique

L'être humain est un système extrêmement complexe. On peut le considérer à travers de nombreux points de vues, tels que physiologiquement, psychologiquement, ou sociologiquement, par exemple. Prendre en compte toutes les facettes est impossible. Modéliser l'humain revient donc à en faire une représentation partielle.

Dans le contexte cyber dans lequel ces travaux se déroulent, les besoins de modélisation de l'être humain sont corrélés à l'objectif que nous nous fixons : évaluer l'être humain à l'aune des vulnérabilités qu'il peut introduire dans un système. Pour cela, il convient d'identifier les propriétés humaines, que l'on appelle facteurs humains, qui peuvent induire une vulnérabilité qui pourra être exploitée dans une attaque cyber.

La notion de facteur humain est large. Il a donc été nécessaire d'une part d'identifier les facteurs humains pouvant avoir une influence sur la vulnérabilité humaine. C'est une tâche ardue puisqu'il faut dans un sens sortir de sa zone de confort et s'attacher à mener une étude de la littérature dans les sciences traitant de l'humain, à savoir les sciences humaines. D'autre part il faut en dériver un moyen de les représenter conceptuellement de manière à les prendre en compte dans une approche d'évaluation du risque de vulnérabilité humaine.

Une fois la capacité de représentation d'un système humain au sein d'une méthode d'ingénierie réalisée, l'autre challenge à relever consistait à évaluer si l'humain pris en compte représente une vulnérabilité pour le système. Pour cela, il faut disposer de moyens d'évaluation, permettant de calculer le risque de vulnérabilité potentielle ainsi que le risque qu'une éventuelle vulnérabilité se propage au système étudié.

Ces travaux se sont déroulés dans le cadre de la chaire de cyberdéfense des systèmes navals [1], qui a financé la thèse de Paul Perrotin, en collaboration étroite avec Naval Group [2].

5.2 Contributions

Les contributions résultantes de ces travaux sont de trois niveaux : le premier porte sur la définition d'un langage permettant de représenter l'humain ; le second sur la fourniture

1. chaire de cyberdéfense des systèmes navals : https://chaire-cyber-navale.fr/
2. Naval Group : https://www.naval-group.com/fr

d'un moyen probabiliste d'estimation de la vulnérabilité d'un humain au sein d'un SoSTS ; enfin, le dernier sur la manière dont la présence d'un opérateur vulnérable peut transformer par propagation un autre opérateur en opérateur vulnérable.

5.2.1 Hos-ML : un langage de modélisation d'architecture SoSTS dans un contexte cyber

Dans le cadre de la problématique consistant à modéliser l'humain, nous avons proposé deux contributions principales : la première consiste en un métamodèle de haut niveau d'abstraction permettant de représenter l'humain à travers le prisme de ses propriétés utiles pour la détection de la vulnérabilité humaine ; la seconde prend la forme du langage de description d'architecture SoSTS, appelé HoS-ML. Il est basé sur une version plus spécifique du métamodèle précédemment cité. Il permet de représenter une architecture décrivant une chaine fonctionnelle composée d'opérateurs humains caractérisés par leurs propriétés.

Dans un premier temps, nous avons défini un métamodèle générique de représentation de l'humain. Pour cela, en nous appuyant sur la littérature, nous avons identifié le besoin de prendre en compte deux types de facteurs humains : les `facteurs directs` et les `facteurs indirects`. Les facteurs humains directs font référence à des propriétés propres à l'humain : la `compétence`, l'`expérience`, la `fiabilité`, la `conscience`, la `confiance`, la `robustesse`, le `niveau informationnel`, la `coopération organisationnelle` et la `stabilité émotionnelle`). Les facteurs humains indirects concernent l'environnement dans lequel l'humain évolue. Nous avons identifié : le `management`, la `politique de sécurité`, la `culture d'entreprise`, la `communication`, l'`exigence de la tâche`, les `ressources` et la `position`. Pour la définition de ces facteurs, nous renvoyons à [Perrotin, 2022]. Afin d'éclaircir les concepts ici présentés, illustrons les par le fait qu'il est évident qu'un opérateur ayant une faible stabilité émotionnelle risque plus d'induire une vulnérabilité dans le système s'il se retrouve dans une situation de stress. De même, la communication associée à un poste peut favoriser l'émergence de vulnérabilité si l'opérateur ne gère pas correctement les communications qui lui sont demandées dans une situation donnée.

D'autre part, les processus humains étant souvent des phénomènes de groupe, ou ayant souvent des impacts sur le groupe, nous avons retenus qu'il était nécessaire de représenter l'humain en tant qu'opérateur humain mais également les autres humains avec lesquels l'opérateur humain modélisé interagit. Le métamodèle proposé permet donc de considérer un humain comme une combinaison de facteurs directs et indirects en relation avec d'autres humains.

Nous avons ensuite dérivé le métamodèle initial en un langage de modélisation. Pour cela, nous avons identifié la liste des facteurs humains à prendre en compte en nous appuyant sur la littérature. Plus précisément, nous nous sommes appuyés sur de nombreuses publications issues des SHS, notamment en psychologie, en sociologie et en sciences du management, pour identifier quels étaient les facteurs humains à retenir. Ce travail transdisciplinaire a été riche et complexe. Il a fallut traduire des théories issues de ces domaines en des propriétés discrètes manipulables par nos modèles.

Les facteurs retenus et leur valeurs possibles ont été intégrés dans le métamodèle de HoS-ML, le langage de modélisation que nous avons développé. Ce langage est inspiré du langage STS-ML [Dalpiaz et al., 2016]. Ce dernier permet la création d'un recueil d'exigences de sécurité pour la conception d'un système socio-techniques. Nous nous sommes inspirés de la vue architecturale de ce langage, pour représenter l'architecture du système à travers les notions de `Rôle` et d'`Acteur`, ainsi que de leurs différents `Objectifs` et `Documents`. La notion de `Rôle` caractérise l'opérateur humain attendu, c'est à dire l'opé-

rateur humain "idéal" pour l'architecte ; la notion d'`Acteur`, l'opérateur humain remplissant un rôle défini (et pouvant ne pas exactement correspondre au `Rôle`). La notion de `Document` recouvre toute information échangée entre deux opérateurs humains dans le contexte du SoSTS (données, messages oraux, numériques ...). La notion d'`Objectif` représente l'objectif a atteindre par le rôle ou par l'acteur. Nous avons repris cette manière de décrire une architecture socio-technique mais en l'adaptant à notre problématique.

Le métamodèle de HoS-ML a été implémenté au sein d'un outil, *Hos-ML Designer*, généré par Sirius. Il permet de représenter des modèles d'architectures de SoSTS et est décrit en détail dans [Perrotin, 2022].

5.2.2 Évaluation de la vulnérabilité humaine : une approche stochastique

Une fois défini et spécifié un moyen de décrire les architectures SoSTS en se focalisant sur la description des facteurs humains des différents opérateurs, nous avons proposé une démarche visant à estimer la probabilité de l'existence d'une (ou de plusieurs) vulnérabilité(s) humaine(s) dans le modèle réalisé. Celle-ci repose sur deux contributions : d'une part, la spécification d'un réseau bayésien permettant de calculer des probabilités de vulnérabilité humaine, et d'autre part la définition d'une démarche visant à s'appuyer sur les probabilités obtenues afin d'estimer la probabilité que cette même vulnérabilité se propage au niveau de l'architecture.

En effet, le langage HoS-ML permet de décrire une architecture SoSTS à travers le profil des opérateurs humains qui la composent et leurs relations. Les modèles réalisés permettent à l'architecte de représenter une architecture centrée sur les opérateurs humains d'une chaine fonctionnelle, et ainsi de raisonner sur les modèles réalisés. Il peut donc s'en servir pour essayer de détecter une vulnérabilité humaine. Cependant, automatiser le processus de détection de la vulnérabilité est préférable, ne serait-ce que pour le passage à l'échelle puisque HoS-ML peut-être utilisé pour représenter des modèles de grandes tailles. Or, cela pose deux problèmes. D'abord, il convient dans un premier temps de définir une méthode de détection de la vulnérabilité humaine. Dans un second temps, il faut trouver un moyen d'évaluer le risque de propagation de la vulnérabilité identifiée au reste du système modélisé. En effet, en fonction de la nature de la vulnérabilité, un opérateur peut éventuellement contaminer un autre, ou a minima les fonctions qu'il peut avoir à réaliser.

Dans le contexte que nous venons de décrire, nous proposons trois contributions étroitement liées. D'une part, nous avons proposé une méthode de détection de la vulnérabilité humaine. D'autre part, nous avons spécifié un réseau bayésien permettant le calcul de la probabilité de l'existence de vulnérabilité humaine. Enfin, nous avons identifié une approche pour évaluer la probabilité de propagation d'une vulnérabilité humaine identifiée au reste du système.

La méthode que nous avons proposée pour estimer une vulnérabilité humaine est assez classique en ingénierie. Elle consiste à définir un standard attendu (ici une valeur donnée pour un facteur humain donné), puis à évaluer les conséquences d'un delta entre le standard attendu et la valeur estimée du système en fonctionnement (ici la valeur du facteur considéré chez l'opérateur humain étudié). Cela revient par exemple à estimer la déviation entre les valeurs des facteurs d'un rôle, qu'on considère comme idéales, et les valeurs des facteurs de l'acteur jouant ce rôle. Se posent plusieurs problèmes : (i) comment déterminer sur la base de ce delta de valeurs qu'une vulnérabilité existe ? (ii) comment généraliser ce résultats alors qu'il est fortement dépendant du contexte dans lequel le SoSTS est amené à fonctionner. En effet, considérons le facteur direct traduisant la résilience au stress chez un pilote d'avion. Il est aisé de voir que le delta existant entre la valeur idéale de ce facteur et la

valeur estimée doit être le plus faible possible à cause des conséquences (crash éventuel) que pourrait engendrer cette situation. On peut éventuellement tolérer de plus grands écarts dans d'autres métiers.

Nous avons donc proposé d'appliquer cette méthode générique à la détection de la vulnérabilité humaine. Un calcul est effectué pour établir la variation de la valeur entre chaque facteur direct du rôle et de l'acteur et d'évaluer par expertise les conséquences de ce delta dans le contexte où l'acteur serait soumis à une attaque cyber.

Cependant, la méthode que nous avons proposée, appliquée par des experts, se confronte à un principe de réalité. Plus le nombre de facteurs directs est important, plus les combinaisons de valeurs sont nombreuses, plus sera potentiellement aléatoire le résultat de l'expertise. Il a donc été nécessaire d'automatiser son application. Pour cela, il a fallut disposer d'un outil calculatoire permettant de jouer efficacement des différentes combinaisons de valeurs de facteurs, tout en tenant compte des éléments contextuels dans lesquels le SoSTS va évoluer.

Nous avons donc proposé d'utiliser un réseau bayésien afin de déterminer les risques de vulnérabilités humaines dans le système. L'intérêt de cette approche c'est d'une part qu'elle détermine une valeur probabilistique. Il sera donc aisé de définir des niveaux de seuils en fonction du contexte. L'autre intérêt c'est qu'à partir du moment où les experts en facteurs humains peuvent définir une cartographie des risques, sous la forme de combinaisons de valeurs amenant à un risque, il est possible de paramétrer le réseau avec. Dans nos travaux, nous avons défini une cartographie standard. Nous l'avons construite en étudiant à la fois la littérature sur le sujet, mais également en lien étroit avec notre partenaire industriel. Cela a abouti à la création d'une table de probabilités prenant en compte les variations entre valeurs des facteurs directs des rôles et celles des acteurs, corrélées avec les valeurs du facteur indirect.

5.2.3 Définition d'une méthode d'estimation du risque de propagation d'une vulnérabilité humaine

Disposant d'un moyen d'évaluer la probabilité d'une vulnérabilité humaine chez un acteur, nous nous sommes intéressés ensuite à évaluer le risque de propagation de cette vulnérabilité à l'ensemble de la chaine fonctionnelle humaine modélisée. En effet, la propagation d'une vulnérabilité humaine au sein d'un SoSTS est l'action qui consiste à transformer un opérateur non vulnérable en opérateur vulnérable du fait de la présence de l'opérateur vulnérable initialement touché dans le SoSTS. 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.

Pour cela, nous avons défini des fonctions calculatoires utilisant plusieurs critères tels que la position hiérarchique ou le niveau de confiance par exemple. Ces fonctions sont appliquées en fonctions des liens existants entre acteurs humains tels que la transmission d'information ou la délégation d'objectifs.

En s'appuyant sur ces principes, nous avons proposé trois méthodes de calcul de la probabilité de propagation d'une vulnérabilité humaine. La première a été construire de manière *ad-hoc* conjointement avec notre partenaire industriel. Pour la deuxième, nous nous sommes attachés à définir la propagation dans le cadre d'attaques informationnelles par fausses informations. Pour la troisième, nous avons fait de même pour la contamination émotionnelle. Il faut voir ces trois méthodes comme des filtres que peut appliquer l'ingénieur en fonction des situations de propagation possibles qu'il veut étudier. Ces trois approches ne sont bien évidemment pas les seules possibles et il sera possible aux ingénieurs d'en

développer de nouvelles.

5.3 Validation

Pour valider notre approche, nous avons choisi une méthodologie par cas d'études [Brereton et al., 2008]. En effet, la validation a été menée en lien étroit avec notre partenaire industriel. Cela nous a permis de confronter notre approche à des cas industriels réels. La méthodologie choisie nous a permis d'encadrer la campagne de validation. Cette dernière a été menée sur 2 cas d'études industriels. Pour chaque cas d'étude, deux scénarios ont été développés. Quatre experts ont été interrogés sur les résultats, deux architectes systèmes seniors et deux experts opérationnels seniors. Le détail de la campagne de validation est consultable dans [Perrotin, 2022]. Les vulnérabilités trouvées en utilisant notre approche ont été validées et jugées crédibles. Une des vulnérabilités trouvées avait d'ailleurs été identifiée récemment par d'autres moyens.

Troisième partie

Chemins de traverse – les modèles dans l'univers de la Défense

Évoluer dans une académie militaire n'est pas neutre. Cela implique une proximité certaine avec les problématiques intéressant la Défense Nationale. Outre certains aspects tels que l'étique personnelle sur lesquels il est naturel de s'interroger à un moment ou à un autre, cela impacte à la fois mes activités d'enseignement et de recherche. Du point de vue de l'enseignement, mes élèves sont les futurs officiers de demain. Concrètement, les élèves de troisième année de l'ESM et ceux de deuxième année de l'EMIA peuvent se retrouver sur des théâtres de guerre un an après être sortis de l'AMSCC. La manière et le contenu de mes enseignements intègrent forcément cet aspect. Ces derniers entrent dans le cadre de la délivrance d'un diplôme d'ingénieur et/ou d'une licence en sciences de l'ingénieur. Mais là où dans d'autres écoles ou universités, le domaine applicatif étudié sera multiple, à AMSCC il sera nécessairement plus orienté vers des applicatifs militaires et/ou plus généralement défense.

D'un point de vue de la recherche, il en va de même. Certaines thématiques peuvent se développer plus naturellement que dans le civil. Cela est dû à la fois au contexte propre à l'Académie mais également aux partenaires institutionnels et industriels avec lesquels nous collaborons sous diverses formes. Ainsi, dans cette partie, j'ai choisi de montrer deux aspects de mes activités de recherche qui n'auraient sans doute pas vu le jours hors de ce contexte particulier. Le premier, que je décris dans le Chapitre 6, concerne une approche d'utilisation de l'IDM dans un contexte opérationnel militaire, et plus spécifiquement dans le cadre de la rédaction et de l'utilisation des ordres d'opérations. Le second, présenté au Chapitre 7 est plus éloigné de mes thématiques habituelles. Il entre néanmoins dans le cadre de la Cyber et plus particulièrement de son volet Champs Immatériels. Il concerne la lutte informatique d'influence (LII) et plus spécifiquement le traitement des Fake News. Il vise à proposer une démarche d'intelligence artificielle explicable pour la détection/génération de Fake News. A la différence des autres travaux présentés dans ce document, ceux mis en lumière dans cette partie ont été réalisés hors du cadre d'encadrement de thèse de doctorat et sont toujours en cours.

Dans les deux cas, on retrouve la prise en compte du facteur humain dans les modèles, à des niveaux différents. Dans le premier, nous avons abordé une problématique future qui est celle du commandement d'un homme, ici l'officier, à un panel de subordonnés pouvant être à la fois humain et techniques (robots, drones ...). Dans le second, nous avons dû comprendre et intégrer dans nos modèles la manière dont les humains réagissaient à la désinformation, ce qui revient à modéliser une partie d'un processus cognitif.

Chapitre 6

L'IDM pour le domaine militaire opérationnel

Contents

Les travaux présentés dans ce chapitre sont directement le fruit de mon intégration à l'AMSCC. Ils constituent une approche exploratoire visant à étudier l'applicabilité d'une démarche de type IDM appliquée au domaine militaire opérationnel. Menés principalement en collaboration avec Jérémy Buisson et plusieurs collègues évoluant en écoles militaires, ils ont conduit à la publication de trois articles [Belloir et al., 2019, Buisson et al., 2020, Belloir et al., 2022a], relatant essentiellement des preuves de concepts.

6.1 Contexte et problématique

Lors de mon arrivée à l'AMSCC, une de mes priorités a été de comprendre le nouvel environnement dans lequel j'allais évoluer : le domaine militaire. Comme tout domaine applicatif, ce dernier a son vocabulaire propre, ses méthodes propres, et ses enjeux propres. Ainsi, afin de monter en compétences sur le sujet, j'ai cherché à établir un lien entre mes activités de recherche passées et ce nouvel environnement. Mon cœur de métier étant l'ingénierie, j'ai donc cherché à tisser un lien entre cette spécialité et ce cadre applicatif. Pour cela, j'ai choisi de procéder par la méthode des petits pas, c'est à dire de mener un certain nombre d'expérimentations à partir d'un point de départ puis de suivre le fil en incrémentant la problématique traitée au gré des opportunités.

Un des intérêts de travailler à l'AMSCC tient dans sa proximité entre chercheurs en SHS et en sciences de l'ingénieur. Dans ce contexte, mêler le concept de "sciences" et "guerre" amène souvent à opposer celui "d'art" sous la forme d'une opposition entre *art de la guerre* et *science de la guerre* en s'appuyant sur les écrits de Sun Tzu [Sun Tzu, v JC] et de Von Clausewitz [Von Clausewitz, 1832]. Si la question, d'après ma compréhension, n'est pas totalement tranchée, on peut observer le travail de théorisation et de formalisation des opérations militaires (citons par exemple le colonel Goya [Goya, 2014] et le général Yakovleff [Yakovleff, 2006]) qui s'approche de la construction de méthodes d'ingénierie empiriques comme on peut les connaître en ingénierie logicielle ou système. En réalité, nombre d'actions dans la conception d'une opération militaire sont proches d'actions que l'on mène en ingénierie. Recueillir de la connaissance sur le sujet à traiter, comprendre l'objectif du donneur d'ordre, analyser avec une vision haut-niveau, puis appliquer une approche descendante s'apparentant à de la conception, évaluer la pertinence du résultat escompté, par fois par simulation, mettre en œuvre et analyser le résultat obtenu en vu d'améliorations futures. En effet, les produits à réaliser qu'on appellera ici des *missions* sont la plupart du temps uniques, rendant *de facto* la phase de test assez spécifique.

Les Armées elles-mêmes définissent des méthodes et des outils spécifiques facilitant la réalisation des actions. On peut citer par exemple la Méthode d'Élaboration d'une Décision Opérationnelle Tactique (MEDOT) [Centre de doctrine d'emploi des forces (CDEF), 2014]. Il s'agit du nom du processus militaire de prise de décision de l'armée de terre. Il consiste en une démarche visant à analyser l'ensemble des tenants d'une situation donnée (objectifs à atteindre, forces en présence, terrain, contexte, etc.), afin de définir le point clé de la manœuvre, l'*effet majeur*. Le parallèle avec une méthode de gestion de projet est assez simple à faire. D'ailleurs, certains processus et formalismes sont structurés de manière à pouvoir être utilisés dans le contexte d'opérations interarmes et/ou interalliées comme celles menées dans le cadre de l'OTAN. Ainsi, le STANAG 2014 [NATO, 2000] décrit la manière dont les ordres d'opérations doivent être rédigés dans une opération OTAN.

Par ailleurs, les systèmes d'information et les systèmes d'armes employés par les armées modernes sont de plus en plus inter-opérants et connectés. On peut citer par exemple le cas du programme Scorpion [1] de l'armée de terre française, qui vise à améliorer le combat collaboratif via un système d'information de combat commun.

Enfin, si on observe les récents évènements en Ukraine, l'utilisation des drones, qu'ils soient autonomes ou non, et leur usage massif dans le processus de combat, forment une tendance massive pour laquelle les doctrines d'une part, et les moyens d'interopérabilité avec les systèmes plus classiques et leurs acteurs d'autre part, doivent évoluer.

On le voit, le domaine militaire offre des problématiques tournant autour de l'interopé-

1. Programme Scorpion : https://www.defense.gouv.fr/terre/nos-materiels-nos-innovations/nos-innovations/dossier-programme-scorpion/programme-scorpion, consulté le 12 juin 2023

rabilité, que ce soit entre systèmes, entre humains ou à l'intersection des deux. Dans ce contexte, nous nous sommes interrogés sur l'apport éventuel des techniques d'IDM afin d'à la fois offrir un moyen d'expression et de représentation par les modèles mais également comme support à l'intégration.

6.2 Contribution

Les travaux décrits ici s'articulent autour de trois contributions principales, pouvant se combiner dans une vue plus large de l'apport de l'IDM au monde militaire, et plus spécifiquement aux actions tactiques menées dans le cadre de l'armée de terre française.

6.2.1 OPORD-ML : un langage de modélisation d'ordres d'opération

Un ordre d'opération (*OPORD*) est un document rédigé par un supérieur à destination de ses subordonnés et décrivant une action future à mettre en place, constituant ainsi la mission à réaliser. En fonction de leur niveau de responsabilité, ces derniers conçoivent leur opération, puis rédigent chacun un autre ordre d'opération à destination de leurs propres subordonnés. Le mouvement se répète jusqu'au niveau de commandement le plus bas, dans une sorte de récursivité dans laquelle chaque ordre d'opération à un rang n se raffinerait à un rang *n-1* jusqu'à atteindre un niveau de granularité spécifique qui interromprait le processus.

Actuellement, la rédaction des ordres d'opération est faite au format textuel, en respectant cependant un format inspiré de la norme OTAN et décrite par le STANAG 2014 [NATO, 2000]. A la manière de la translation qui a été opérée en ingénierie des approches centrées document au profit des approches centrées modèle, nous avons proposé dans [Belloir et al., 2019] d'utiliser les apports de l'IDM pour la rédaction des ordres d'opération. Nous avons donc développé un langage de rédaction d'ordres d'opération basé sur une syntaxe abstraite exprimée à l'aide d'un métamodèle inspiré du STANAG 2014 et d'une syntaxe concrète, et de son environnement outillé, généré à l'aide du logiciel SIRIUS [2] construit sur Eclipse. L'intérêt de cette solution est de permettre de bénéficier des apports des modèles (représentativité visuelle, réutilisabalité, traçabilité, lien avec des simulateurs ...) là où l'utilisation d'une approche documentaire implique pour les mêmes propriétés des efforts non négligeables et un risque accru d'erreurs liées à la manipulation des informations au sein des documents. L'autre apport vient de la possibilité de traiter cet ordre par une machine de manière autonome et automatique.

6.2.2 Vers une utilisation de l'IDM en support aux centres d'opérations

La rédaction des ordres d'opération n'est que le point final d'un processus conceptuel. Ce dernier est généralement le fruit d'une planification réalisée par un état major dans un centre d'opérations (CO). Actuellement ce processus est normalisé et majoritairement mis en œuvre à l'aide de séances de brainstorming utilisant des vue de type présentations et du langage naturel. Si l'on peut comprendre la facilité qui guide ces approches, il conviendra dans le futur de passer de l'utilisation de tels formalismes à des approches centrées modèles. En effet, l'apport majeur de ces derniers est d'être interprétable par des machines

2. SIRIUS : https://www.eclipse.org/sirius/, consulté le 12 juin 2023

et notamment par des outils de simulation et d'aide à la décision. En cela, le processus de conception d'une opération est très proche de celui d'un système de systèmes.

Ainsi, dans [Buisson et al., 2020], nous proposons d'utiliser les techniques et outils issues de l'ingénierie des SoS de manière à construire les méthodes, techniques et outils de la future génération de CO. Plus particulièrement, nous identifions un certain nombre de challenges visant à soutenir la création de CO de nouvelle génération dans lequel la conception des opérations préalables à la rédaction des ordres d'opération s'appuie sur de telles technologies. Parmi ces challenges, sont abordées des problématiques telles que l'évolutivité, la modélisation collaborative, l'agilité, la modélisation multi-domaines. De plus, nous soutenons que l'ajout de techniques d'intelligence artificielle pourrait améliorer la capacité des centres d'opérations à mieux prévoir les mouvements ennemis notamment en modélisant les doctrines ennemies. Du point de vue de la modélisation des capacités amies, d'autres challenges apparaissent notamment dans des champs tels que la modélisation des ressources, ou la capacité à être non prédictif pour contrer l'ennemi. Enfin, la mise en place d'une telle approche par modèles, pour être efficiente, doit également bénéficier d'un lien numérique avec les unités opérationnelles dans un continuum cybernétique. En effet, il s'agit de lier des capacités de traitement automatique des ordres d'opération, pour les unités autonomes notamment, avec des capacités cyber défensives, voire offensives.

6.2.3 Vers une section hybrides : l'IDM comme passerelle

Dans la section précédente, nous avons proposé de construire les outils et méthodes nécessaires au fonctionnement d'un centre d'opérations en s'appuyant sur les outils et les méthodes issues de l'ingénierie des SoS. Parmi ceux-ci, l'IDM, en utilisant les modèles comme pierre angulaire, est candidate comme support méthodologique et technologique.

Un des reproches souvent faits aux approches basées sur l'IDM est la complexité de cette dernière, et la difficulté à être prise en main. Or, dans le contexte militaire, s'il est aisé de faire le parallèle entre un ingénieur et un officier dans leur rôle conceptuel, dans leur rôle opérationnel, ce n'est plus le cas. L'officier doit être efficient, réactif et les outils qui doivent l'aider doivent également l'épargner d'une augmentation de sa charge cognitive. Il faut donc convaincre que l'utilisation des modèles n'est pas synonyme de complexité supplémentaire.

Aussi, dans [Belloir et al., 2022a], nous avons mené une expérimentation dont l'objectif était d'illustrer le fait que des outils basés sur les modèles pouvaient amener une plus-value sans ajouter de complexité. Dans ce contexte, nous nous sommes placés dans le cadre d'un officier sur le terrain qui aurait à commander une section hybride composée d'humains et de robots. Pour cela, nous avons modélisé dans un méta-modèle la doctrine de combat PROTERRE [Armée de Terre Française, 2014] qui décrit les actions élémentaires d'une section au combat : *reconnaitre*, *tenir*, *soutenir*, ...). Parmi la manière de donner l'ordre d'effectuer ces actions élémentaires, la doctrine décrit un langage gestuel. Ce langage gestuel peut-être vu comme une syntaxe concrète. Nous avons donc développé un gant connecté que devrait porter le chef de section, et nous avons montré qu'en utilisant une approche IDM, le chef pouvait donner un ordre gestuel, reconnu automatiquement par un composant entrainé par apprentissage, qui l'associait à des ordres décrits dans le métamodèle, avant de le transmettre automatiquement aux robots de la section selon une autre syntaxe concrète.

6.3 Validation

Les travaux présentés dans cette partie et plus particulièrement dans ce chapitre se sont déroulés dans un contexte un peu différent de ceux présentés dans le reste du manuscrit.

En effet, ils ont été menés lors de collaborations avec un certain nombre de collègues et soutenus essentiellement par des projets étudiants. En cela, la partie validation relève plus de la preuve de concept que de la preuve de validité, si l'on peut l'appeler ainsi. Nous avons donc développé un éditeur de modélisation des ordres d'opération implémentant un langage textuel pour la première contribution. Celui-ci est très imparfait et pose quelques questions quant à l'utilisabilité d'un tel langage dans un cadre de fort stress et de saturation cognitive comme l'est le combat. Néanmoins il montre la faisabilité de l'approche.

La deuxième contribution relève de l'article de position. En cela, il n'y a pas de validation associée, sinon une validation théorique de l'approche proposée par Stéphane Taillat, un historien spécialiste du monde militaire et chercheur à l'AMSCC.

Enfin, la troisième contribution a bénéficié d'une expérimentation plus poussée puisqu'elle a conduit au développement d'un gant connecté et à la réalisation d'un cas d'étude sur un robot simplifié. Là encore, nous sommes plus dans la preuve de concept que dans la validation.

Ces preuves de concept méritent d'être approfondies. En effet, la validation des systèmes techniques dans un domaine applicatif tel que la Défense est particulièrement fondamental. Ces derniers sont souvent des systèmes critiques et leurs bonnes utilisations, et inversement leurs défaillances, impliquent souvent des conséquences dramatiques. Les conditions d'utilisation de tels systèmes en situation de combat réel forment un environnement d'exploitation particulièrement intense et riche en évènements émergents pouvant les perturber. A titre d'exemple, citons le cas de la première utilisation des missiles MdCN français lors de l'opération Hamilton au large de la Syrie qui a connu un double échec de mise à feu [3] alors que les validations, à la fois par l'industriel, la DGA et la Marine avaient été réalisées avec succès, mais hors contexte opérationnel réel. On mesure donc l'importance du neuvième niveau de TRL [Mankins, 1995] qui est celui d'une validation opérationnelle en conditions réelles. Sans être à ce niveau, cela illustre bien la nécessité de valider chaque niveau de TRL et donc de conduire une validation de nos travaux sur ce domaine.

3. Lors de l'opération Hamilton, 3 frégates françaises étaient sur zone. Elle devaient tirer des missiles de nouvelles génération. Le tir des deux premières frégates n'a pas été possible. La dernière, qui était en réserve, a réussi sur la toute fin de la fenêtre de tir, à la limite donc de l'échec de la mission : https://www.opex360.com/2018/11/03/syrie-marine-nationale-a-tire-les-enseignements-des-rates-de-loperation-hamilton/, consulté le 30 mai 2023

Chapitre 7

Une approche conceptuelle pour les Fakes News

Contents

Les travaux présentés dans ce chapitre entrent dans le champ de la lutte informatique d'influence (LII). Ils proposent une approche conceptuelle visant à mieux définir ce qu'est une Fake News. Ils proposent une utilisation de ce modèle en tant que pierre angulaire dans une démarche d'explicablité des résultats fournis par des outils d'intelligence artificielle dans des processus automatiques de génération et/ou de détection. Ils ont été mené en collaboration avec Wassila Ouerdane, maitresse de conférences à Centrale SupElec et Oscar Pastor professeur à l'Université Polytechnique de Valence. Ils ont conduit à la publication de [Belloir et al., 2022c, Belloir et al., 2022b].

7.1 Contexte et problématique

Depuis quelques années, la désinformation est un sujet particulièrement impactant pour les sociétés modernes. Les exemples sont nombreux et multiples et ciblent des sphères aussi diverses que la sphère politique, économique ou médicale. On peut citer les actions de manipulation lors des élections présidentielles américaines ou françaises, les attaques réputationelles visant à décrédibiliser des entreprises ou encore les campagnes anti-vaccins qui ont atteint un paroxysme lors de la pandémie de Covid. Dans le cadre des actions diplomatiques ou militaires, récemment la France a subi des attaques informationnelles dont les résultats ont été de perdre toute forme d'influence politique dans certains pays. Cela a notamment conduit au retrait des forces françaises du Mali et à la fin de l'opération Barkhane alors que les opérations militaires étaient globalement couronnées de succès. Le général Burkhard, CEMA, a déclaré le premier septembre 2022 à ce propos : "il est indispensable de développer nos capacités d'influence pour gagner la guerre avant la guerre". En lien, le Président de la République a inclus cette capacité dans la Revue Nationale Stratégique le 9 novembre 2022, la déclarant comme "fonction stratégique".

Les artéfacts et actions utilisés dans le cadre de LII sont pluriels. Parmi ceux-ci, l'utilisation de Fake News est particulièrement importante. Malgré cela, l'utilisation du terme Fake News est parfois employé de manière incorrecte. Certaines définitions du terme sont parfois d'ailleurs contradictoires. Il y a donc un besoin à clairement caractériser ce qu'est une Fake News.

Par ailleurs, de nombreux travaux portent sur la détection de Fake News ; moins sur la génération, mais il y en a. Globalement, nombres d'approches utilisent des outils mathématiques ou des outils d'intelligence artificielle pour mettre en œuvre cette détection. L'apprentissage de ces derniers est souvent basé sur des jeux de données plutôt thématiques, rendant la détection par induction thématique, donc limitée. De part l'aspect boite noire des outils utilisés, il peut-être aisément opposé un doute aux résultats amenés par ces approches. En effet, dans les milieux complotistes parfois très technophobes, demander à se fier à une intelligence artificielle opaque n'est pas la meilleure manière de convaincre. Nous pensons qu'une autre voie est possible.

Dans ce contexte, nous avons développé un axe de recherche reposant sur une proposition originale. Il s'agit de caractériser précisément ce qu'est une Fake News à l'aide d'un modèle conceptuel. L'idée que nous proposons est, une fois le modèle conceptuel défini, d'utiliser celui-ci comme pierre angulaire pour la détection de Fake News. En effet, le modèle peut-être considéré comme une sorte de canevas étalon : si une information qu'on souhaite évaluer est conforme au modèle, il s'agit alors d'une Fake News.

Le travail présenté ici entre dans le cadre de la LII. Il a été mené en collaboration avec Wassila Ouerdane, maîtresse de conférence à Centrale SupElec, et Oscar Pastor, professeur à l'Université Polytechnique de Valencia. Il a également impliqué un certain nombre d'élèves-officiers de l'AMSCC.

7.2 Contributions

Ce travail a conduit à des contributions en deux temps. Dans un premier temps, nous avons mené une étude visant à identifier les concepts clés permettant de caractériser une Fake News. Cette première étape a mené à la publication d'un article [Belloir et al., 2022c]. Ces travaux sont décrits en Section 7.2.1. Dans un deuxième temps, nous avons défini un modèle conceptuel intégrant les points identifiés en amont et l'avons publié dans [Belloir et al., 2022b]. Nous présentons ce travail en Section 7.2.2.

7.2.1 Définition et caractérisation du concept de Fake News

L'objectif de proposer une caractérisation des Fake News est d'identifier en détail quelles sont les éléments pertinents qui caractérisent conceptuellement les différentes dimensions à prendre en compte pour traiter correctement les données d'une Fake News. Pour cela, nous avons dans un premier temps chercher à classifier les différents types d'informations (au sens nouvelles). Nous défendons l'idée qu'une Fake News est avant tout une information visant à altérer la vérité avec une intention de nuire. Notre première contribution a donc été de proposer la classification présenté dans la Figure 7.1.

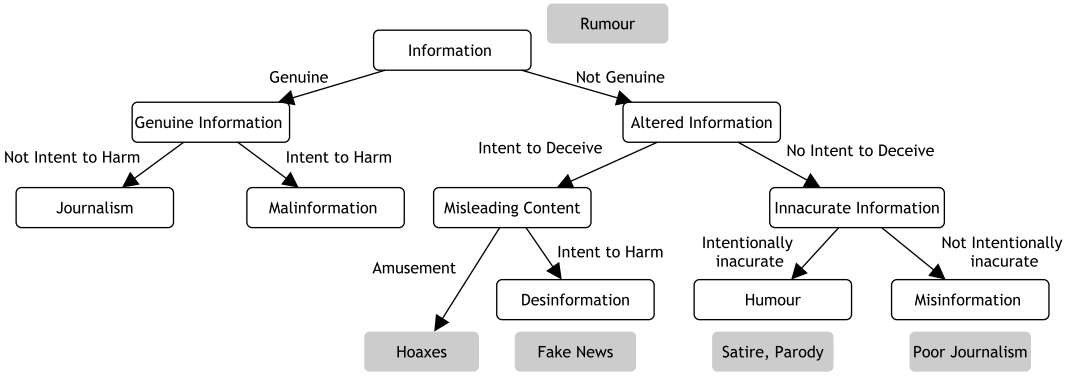


FIGURE 7.1 – Classification de l'information : Fake News et des concepts proches [Belloir et al., 2022c]

La seconde contribution de ce travail a été d'identifier les grands concepts propres à la notion de Fake News. Pour cela, nous avons procédé de manière empirique en menant une étude d'observation de Fake News existantes. Cela nous a conduit à identifier quatre propriétés d'une Fake News.

Premièrement, une Fake News est la plupart du temps produite dans le cadre d'une campagne de désinformation dont l'organisation est le plus souvent multi-niveaux, répondant à des objectifs stratégiques, opérationnels et tactiques. En cela, on peut dire qu'une Fake News peut-être comparée à une action tactique. Prendre en compte ces trois niveaux permet d'aider à identifier précisément le contexte dans laquelle la Fake News intervient, aidant ainsi à la positionner dans l'actualité. Par exemple, rappelons que 65% des fausses informations sur la vaccination, lors de la pandémie de Covid-19, sont issus de seulement douze personnes [CCDH, 2021]. Cela montre clairement une action coordonnée menée par un petit groupe de personnes que l'on a réussi à clairement identifier.

Deuxièmement, une Fake News cible une catégorie ou un groupe de personnes précis et utilise un processus cognitif basé sur les émotions pour toucher cette dernière. En effet, les Fake News sont construites de manière à déclencher une charge émotionnelle. Cette dernière a pour objectif d'amener le lecteur à tirer une conclusion sous le coup de l'émotion, c'est à dire sans prendre le temps de réfléchir au propos porté par la Fake News. Pour cela, la charge émotionnelle est forcément construite en fonction de la catégorie de personnes ciblée, de manière à maximiser les effets.

Troisièmement, une Fake News est une distorsion de la réalité. Cette distorsion peut d'ailleurs prendre plusieurs formes allant de la fausse information créée de toute pièce à la combinaison d'évènements réels mis ensemble de manière à créer l'impression d'une corrélation alors qu'ils sont complètement indépendants. Dans le premier cas, la fausse

information totalement créée est quand même mis en parallèle avec une vraie information pour assoir la crédibilité de l'ensemble. La distorsion entre le fait vrai et le fait faux mis en relation est d'ailleurs un élément important sur lequel les créateurs de Fake News jouent. En effet, certaines cibles ne nécessitent pas de finesse alors que d'autres au contraire en ont besoin.

Quatrièmement, toujours de manière à répondre à ce besoin de crédibilité, une Fake News s'appuie souvent sur une autorité. Cette dernière peut prendre différentes formes comme par exemple celle d'un "expert", c'est à dire une personne pouvant jouer d'un statut ou d'une expérience particulière de manière à renforcer la crédibilité de l'information véhiculé. Un expert d'un sujet accréditant rarement une erreur, il pourra être utilisé un expert d'un sujet A pour jouer l'autorité sur un sujet B par exemple. Il pourra également être attribué un faux témoignage à un expert réel. Il pourra également être utilisé de faux experts.

Le travail d'étude et de caractérisation des Fake News nous a mené à proposer notre propre définition d'une Fake News : "*Une Fake News est une nouvelle fausse mais vérifiable, composée de faux faits basés sur des faits réels. Rédigée de manière à déclencher une charge émotionnelle, elle vise à tromper ses lecteurs et à influencer leur opinion par une conclusion implicite*".

Ces trois premières contributions sont présentées dans [Belloir et al., 2022c].

7.2.2 Définition d'un modèle conceptuel de Fake News

En nous appuyant sur le travail précédent, nous avons défini un modèle conceptuel complet. Ce dernier est spécifié à l'aide du langage UML. Il définit une classe principale représentant la Fake News. Puis, il s'articule en différentes zones, dans lesquelles se retrouvent les concepts clés identifiés dans la section précédente. On trouve en particulier une zone du modèle qui permet de décrire une Fake News d'un point de vue "*structurel*", c'est-à-dire les concepts caractérisant sa construction à travers les informations qu'elle donne. Elle répond à la question "*quoi ?*". Une seconde zone décrit le coté "sémantique" de la Fake News. Il faut comprendre ici que nous représentons l'effet que cherche à obtenir la Fake News vis-à-vis de ceux qui la consulte. Elle répond à la question "*comment ?*" et "*pour quoi faire ?*". Enfin, un troisième volet vient décrire le contexte opérationnel dans lequel la Fake News est créée, c'est-à-dire les différentes niveaux (tactique, opérationnel et stratégiques) qui sont à l'œuvre ; la Fake News est une action tactique qui se déroule dans le contexte d'une campagne opérationnelle de désinformation elle-même issue d'une action de guerre informationnelle a portée stratégique. Elle répond à la question "*qui ?*".

Nous ne détaillons pas plus ici le modèle et renvoyons à l'article [Belloir et al., 2022b] le présentant en détail en Annexes.

7.3 Validation

La validation de notre approche est encore partielle car ce travail est toujours en cours. En effet, le modèle conceptuel proposé a été évalué par expérimentation. Nous l'avons confronté à plusieurs dizaines de Fake News de différentes formes, cela afin de vérifier sa conformité. Dans l'ensemble, c'est le cas. Il ressort cependant qu'un certain nombre d'éléments du modèle peuvent être difficiles à identifier sur le seul contenu d'une Fake News. Par exemple, identifier le cadre de la campagne de désinformation dans laquelle évolue la Fake News est parfois impossible à un instant t. Il faut parfois combiner plusieurs Fake News ensembles pour voir apparaître ces informations. Plus globalement, une Fake News a sa propre vie : elle nait, vit, se propage et meurt au bout d'un certain temps. Le modèle

capture bien sa partie structurelle, mais peu les éléments de sa vie, et ce sera certainement une évolution future.

Par ailleurs, nous avons également utilisé le modèle pour créer des Fake News, en utilisant un processus que nous avons défini mais pas encore publié. Cela s'est avéré concluant. L'étape suivante sera d'automatiser le processus.

Enfin, nous avons commencé à expérimenté le modèle dans une démarche d'XAI [1] telle que celle prônée par Spreeuwenberg [Spreeuwenberg, 2019]. L'idée originale que nous portons est celle d'utiliser le modèle conceptuel comme élément d'explicabilité des résultats d'une chaîne d'outils d'IA visant à identifier des Fake News. Ce travail, qui nous l'espérons aidera à valider notre proposition, est en cours. Nous avons obtenu un financement de thèse sur le sujet et Angélique Yameogo a débuté ses travaux de thèse le premier décembre 2022 dessus.

1. XAI : Explainable Artificial Intelligence : champ de l'intelligence artificielle visant à permettre à ses utilisateurs de se fier aux résultats produits par des outils basés IA.

Quatrième partie

Au bout des chemins – Conclusion et perspectives

Dans cette partie, nous concluons ce mémoire, en dressant dans un premier temps une conclusion sur le travail de recherche mené, puis, dans un second temps, en dressant des perspectives sur les directions que je souhaite prendre dans les années à venir.

Chapitre 8

Conclusion

Contents

Dans ce chapitre je dresse quelques conclusions sur mon activité de recherche qui est présentée dans ce mémoire. Je récapitule les différents axes de recherche présentés, les doctorants et étudiants encadrés et analyse les résultats de ces recherches sur un axe qualitatif.

8.1 Bilan général

Ce chapitre dresse un bilan des travaux présentés dans ce mémoire. Mon domaine de recherche se situe clairement dans le domaine applicatif de l'IDM. Il vise à intégrer l'utilisation des modèles dans les méthodes de conception, la plupart du temps industrielles. Ma vision est que, dans ce contexte, tout artéfact conceptuel envisagé (logiciel, système, humain, ordre ...) peut-être considéré et modélisé comme un sous-système constituant et le système final comme un système de systèmes.

Dans ce manuscrit, j'ai choisi de montrer trois axes de mes travaux. Le premier s'intéresse au domaine de la conception de l'architecture des systèmes et des systèmes de systèmes. J'y ai développé dans un premier temps une approche visant à permettre une formalisation des exigences pouvant être relâchées en s'appuyant sur le langage Relax. Dans un second temps, j'ai présenté notre approche proposant d'utiliser le paradigme "mission" comme invariant, cela afin de formaliser une architecture abstraite visant à conserver la mission d'un système de systèmes au gré de ses évolutions.

J'ai ensuite montré la manière dont nous appliquions l'IDM dans des démarche de sécurisation par construction. En premier lieu, j'ai présenté une méthode visant à tisser un pont entre les architectes systèmes et les experts en sécurité via l'utilisation de modèles, cela en réifiant le concept "d'asset". Puis, dans un second temps, j'ai décrit nos travaux visant à prendre en compte la vulnérabilité humaine lors de la modélisation de l'architecture des SoSTS.

Enfin, j'ai présenté deux travaux plus spécifiquement liés au domaine de la Défense. Le premier vise à intégrer une approche IDM qui utiliserait les modèles comme pierre angulaire à un système de commandement intégré, allant des centres d'opérations jusqu'à des sections hybrides. Le second utilise les modèles afin de spécifier le concept de Fake News, de manière à permettre de mieux caractériser ces dernières dans un objectif de détection et d'explicabilité.

De nombreux chercheurs creusent leur sillon en profondeur. J'ai préféré creuser le mien en largeur. Cette approche nécessite d'être adaptable, de savoir se remettre en question et de pouvoir s'enrichir des nouveaux domaines qu'on aborde. Oui, concevoir du logiciel n'est pas la même chose que des systèmes physiques ou des comportements humains. Cela étant, je crois avoir montré que, malgré les différences, il est possible d'utiliser des outils similaires pour le faire. Cette adaptabilité se retrouve également dans mon déroulé de carrière.

Par essence, mon approche est avant tout transdisciplinaire. Elle colle avec mon appétence à travailler avec des personnes différentes, issues de domaines et de milieux différents, comme l'atteste la liste de mes co-auteurs, nombreux, sans qui ce mémoire n'aurait sans doute pas vu le jour. Pour moi, faire de la recherche est avant tout une aventure humaine, de partage et d'enrichissement mutuel.

8.2 Regard qualitatif

Dresser un bilan qualitatif est un travail qui nécessite de prendre du recul et d'accepter sa propre autocritique. Aussi, à travers les 4 indicateurs que sont les publications, le travail d'encadrement, les collaborations industrielles que l'on peut avoir, et enfin la démarche de validation, j'essaie ici de dresser un "Polaroïd" de mon activité de recherche.

En premier lieu, on peut voir que mon activité de recherche durant mes années paloises est décevante. Focalisée sur l'enseignement, dans un environnement difficile et contraint, je n'ai pas su me relancer après ma thèse. Le changement s'amorce lorsque j'ai l'opportunité de participer à l'encadrement de la thèse d'Ahmad Manzoor. S'en suit mon détachement

à l'AMSCC, à partir duquel, je monte progressivement en puissance en recherche, tant sur le niveau publications (nombre et niveau), de l'encadrement et du montage de partenariats industriels, alors même que ma charge d'enseignement explose.

8.2.1 Publications

Le bilan des publications est un indicateur parmi d'autres de notre activité de chercheur. Comme tout métrique, il ne vaut que l'importance qu'on veut bien lui accorder. Dans la mesure du possible, je m'attache à évaluer la qualité de mes publications à l'international en fonction du classement CORE australien. Ce choix est discutable et ne résout pas tout, notre communauté scientifique n'ayant pas réussi à atteindre de consensus sur cette question. Toutefois, je le juge relativement pertinent et honnête.

Quand je regarde ce bilan personnel, j'y vois le reflet de ma carrière, avec trois périodes distinctes qui s'y retrouve : la période de la thèse, plutôt active, avec notamment la dynamique apportée par la participation active à un projet européen ; mon début de carrière à l'UPPA, durant lequel on peut constater une dynamique faible, surtout en terme qualitatif ; et enfin, depuis 2013, une dynamique de montée en qualité et en nombre, qu'on peut corréler avec ma réorientation thématique, dans un premier temps en collaboration avec l'IRIT, puis avec mon arrivée en Bretagne.

Le bilan ci-dessous est donné à partir de 2005, date de ma prise de fonction. Pour le détail, je renvoie à la liste donnée en Annexes.

Récapitulatif	International(e)	National(e)	Total
Article en revue	3	2	5
Chapitre de livre	0	1	1
Article en conférence	13	5	18
Article en atelier	5	2	7
Rapport scientifique	1	0	1

TABLE 8.1 – Bilan des publications à partir de 2005

8.2.2 Encadrements

On dit souvent que les étudiants, qu'ils soient en master ou en thèse, sont les petites mains de la recherche. Si cette vision n'est pas totalement fausse (ils nous permettent en effet d'avancer dans nos activités quand le temps de travail pour effectuer nous-même de la recherche est limité), elle est à mon sens réductrice. En effet, on peut imaginer alors que les encadrants sont des sortes de grands donneurs d'ordre qu'appliquent les étudiants. Je ne souscris pas à cela. je vois plutôt mon rôle en tant qu'enseignant-chercheur titulaire et encadrant de projets, quels qu'ils soient, comme un rôle de transmetteur, de guide. Ce rôle est fondamental et rien n'est plus satisfaisant que de voir un étudiant qu'on a guidé devenir de plus en plus autonome.

Depuis mon recrutement en tant que maître de conférences, j'ai participé à l'encadrement de cinq thèses soutenues, j'ai actuellement deux thèses en cours d'avancement et une en lancement. Enfin, une de mes doctorantes a abandonné. Cet abandon a été marquant pour moi. Il a façonné la manière dont j'ai encadré par la suite ; il m'a amené à redoubler d'attention sur l'aspect humain du suivi de thèse. Sur les cinq thèses soutenues, Youssef Ridène est devenu manager logiciel senior chez Amazon après avoir été chef de produit au sein d'une équipe de R&D chez un éditeur de logiciel spécialisé en IDM ; Manzour

Ahmad est enseignant-chercheur contractuel en CDI à l'UPPA ; Nan Messe est maîtresse de conférences à l'Université Toulouse Jean Jaurès ; Imane Cherfa est maîtresse de conférences à l'Université de Blida I ; et finalement Paul Perrotin est chercheur à la DGA-MI.

J'ai également encadré trois projets de recherche pour des étudiants en Master 2 orienté recherche (ou DEA). Dans l'ensemble, j'essaie de faire en sorte de participer à l'encadrement des étudiants de l'AMSCC que nous envoyons en stage international de fin de scolarité. C'est d'ailleurs de là qu'ont commencés mes travaux sur les Fake News. Enfin, j'ai intégré des projets étudiants divers et variés dans mes activités de recherche.

8.2.3 Collaborations industrielles et institutionnelles

Depuis mon recrutement, mes collaborations industrielles et institutionnelles se sont presque toujours déroulées dans le cadre d'un financement de thèse ou de stage de master. Une exception est à noter : la participation au bureau de l'association SysML-France pour laquelle nous organisions des retours d'expérience au profit des industriels. Pour le reste, on peut dresser le même constat que celui déjà exprimé au cours de ce document : même si j'ai pu bénéficier de quelques collaborations industrielles et institutionnelles en amont, l'environnement breton m'a permis de sérieusement développer ces partenariats, notamment auprès des industriels et des institutions de la défense, avec des liens privilégiés tels que le ministère des armées, la DGA-MI, Naval Group et Thales. C'est d'ailleurs en lien avec ce dernier que j'ai pu participer au montage de la chaire de recherche "Cyberdéfense - Cybersécurité St Cyr Thales" dont je suis le titulaire depuis le premier janvier 2023.

D'autre part, depuis 2016, j'ai réussi à trouver systématiquement des financements pour conduire mes recherches. C'est là encore un changement majeur vis-à-vis de ma période paloise. Il y a évidemment une raison contextuelle et géographique, mais cela a nécessité aussi de ma part une agilité sur mes thématiques de recherche et une capacité à porter de nouveaux sujets, comme le montre ce manuscrit.

8.2.4 Validation des approches proposées

Un dernier point que je souhaite mettre en exergue ici concerne la qualité des validations menées dans le cadre de mes travaux. Il en est certaines que j'aurais aimé pousser plus en avant. En effet, les cas d'études permettent de montrer que l'approche fonctionne. Mais ils ne prouvent ni que l'approche est complètement reproductive, ni qu'elle amène une plus-value significative. Dans certains travaux, il aurait été intéressant de présenter l'approche a minima à des spécialistes pour qu'ils l'analysent et donnent des éléments d'évaluation plus pertinents.

Plus généralement, en ingénierie logicielle et/ou système, ou tout au moins dans le sous-domaine qui est le mien, il existe une difficulté de validation qui est récurrente. Mes travaux visent à produire des méthodes basées sur la réalisation langages de modélisation et des processus les utilisant. Nous argons souvent que ces modèles permettent d'être plus efficient, de mieux prendre en compte certains aspects. Évaluer de telles affirmations est complexe car la modélisation de tel ou tel système prend du temps, se réalise souvent en équipe, dépend de l'expérience des ingénieurs la réalisant ... bref se fait dans un cadre humain et est expérimental. Or, recréer ce cadre humain dans un contexte expérimental est difficile et pose des questions légitimes : à partir de combien de modèles réalisés peut-on affirmer que notre assertion est vraie ? A qui demander de participer aux expérimentations ? Des ingénieurs experts ? Comment rémunérer leur travail ? A des ingénieurs juniors (souvent des étudiants en fin de cycle) ? Comment s'assurer qu'ils ont le recul nécessaire à tirer les bonnes conclusions ? Des solutions existent. Nous en avons mises en place dans des travaux

décrits ici, mais si elles améliorent un peu la validation, elles ne sont pas toujours suffisantes à considérer l'assertion comme vraie. Cela reste pour moi une question en suspens, mais pour laquelle, comme pour les points cités précédemment, je souhaite vivement continuer à m'améliorer.

Chapitre 9

Perspectives et projet de recherche

Contents

Dans ce chapitre je propose un certain nombre de perspectives valant projet de recherche. Ce projet s'articule autour de trois axes qui s'appuient à la fois sur l'expertise que j'ai développée et présentée dans ce document, mais également sur des problématiques ouvertes. Les trois axes sont : l'ingénierie pour la cybersécurité, la prise en compte de l'humain dans les processus d'ingénierie, et l'apport de l'ingénierie dans le domaine opérationnel militaire.

En recherche, résoudre des problèmes et répondre à des questions de recherche amène toujours à en ouvrir de nouvelles. Ainsi, mes perspectives sont une continuité directe des travaux que j'ai menés en amont. C'est pour cela que ce chapitre est organisé en trois sections, chacune se voulant une évolution possible, voire déjà amorcée, des différents travaux présentés dans ce mémoire. Par ailleurs, comme le montrent les perspectives décrites ici, la part des travaux envisagés dans le monde de la cybersécurité s'accroit, sans pour autant oublier l'ingénierie qui va avec.

9.1 Ingénierie pour la sécurité des SoS

La compréhension de la problématique de la cybersécurité dans le monde industriel est en train de changer et c'est une bonne chose. Cependant, il est clair que l'intégration de cette dernière dans les processus d'ingénierie n'est pas encore bien prise en compte. Le rapport annuel de l'ANSSI présente un constat inquiétant. En effet, parmi les dix vulnérabilités les plus utilisées dans le cadre de cyber-attaques en 2022, il dit que : " certaines [...] sont corrigées depuis 2021 " [ANSSI, 2023]. Cela montre, que, non seulement il reste des trous dans la raquette, mais également que du seul point de vue de la maintenance applicative et du traitement des vulnérabilités connues, le minimum n'est pas fait. Aussi, est-il nécessaire de continuer à intégrer le risque Cyber dans les processus d'ingénierie. Par ailleurs, la dimension SoS ajoute aux problématiques de cybesécurité un niveau de risque supplémentaire. En effet, une caractéristique distinctive des SoS est l'indépendance managériale de leurs CS, causant une absence de spécifications fixes et la fréquente cohabitation entre CS patrimoniaux et CS récents. L'indépendance managériale des CS limite de plus la capacité du SoS à imposer des mesures de sécurité aux CS. Et l'indépendance opérationnelle des CS autorise des interactions indépendamment du SoS. Quand bien même les propriétés de sécurité de chaque CS seraient bien définies, l'incertitude du comportement global de sécurité du SoS reste un défi ouvert [Hachem et al., 2020, Kopetz et al., 2015]. En effet, le risque de voir apparaître des comportements émergents non désirés est réel. Ainsi, dans les perspectives de cette section, je centrerai mes problématiques dans le cadre spécifique des SoS. Dans ce contexte, je prévois de développer trois axes d'efforts, dont deux sont une continuité directe de mes travaux précédents et le dernier une nouvelle problématique.

9.1.1 Renforcer le lien entre experts en sécurité et architectes

Dans des travaux précédents [Messe et al., 2020a, Messe et al., 2020b], nous avons montré que l'identification des actifs permettait d'améliorer le processus de modélisation de la menace lors de la conception architecturale, en manipulant les aspects de sécurité à plus haut niveau d'abstraction. Par ailleurs, les graphes d'attaque [Zeng et al., 2019] et les arbres d'attaque/défense [Tøndel et al., 2010] permettent de modéliser les menaces en se concentrant sur les motivations des attaquants et leurs impacts sur le système [Naouar et al., 2021].

Être capable de concevoir la sécurité du SoS et d'en prédire les potentielles attaques est une nécessité dans l'environnement de cyber-menaces actuel, compte-tenu de la criticité des domaines applicatifs. Parmi les défis liés à l'ingénierie de la sécurité des SoS, les travaux proposés se concentreront sur la modélisation et l'analyse des architectures sécurisées des SoS en prenant en compte les caractéristiques spécifiques des SoS. Il s'agit de déterminer comment les approches de l'ingénierie logicielle peuvent être étendues pour analyser diverses alternatives d'architectures sécurisées de SoS afin d'évaluer les comportements de sécurité, intrinsèquement émergents dans le contexte SoS, ainsi que les potentielles cyber-attaques, au plus tôt dans le cycle de vie du SoS, lors de la phase de la conception architecturale.

L'évaluation des cyber-attaques à un stade précoce permettra leur catégorisation, leur priorisation, et leur résolution avec des coûts et des délais réduits, en assurant la protection du SoS contre leurs dommages. Plus particulièrement, nous étudierons la manière dont les concepts de sécurités tels que la menace et les attaques pourront être intégrés dans le processus d'ingénierie/de modélisation système, en se concentrant sur les motivations des attaquants. D'autre part, nous chercherons à analyser l'architecture sécurisée d'un SoS afin d'étudier les potentielles cyber-attaques émergentes, de déterminer les éléments architecturaux affectés par chaque action de l'adversaire, et analyser quantitativement et qualitativement cet impact afin de guider l'architecte en lui proposant des solutions de remédiation.

Ce sujet de recherche a reçu le soutien du Pôle d'Excellence Cyber (PEC) à travers le financement de la thèse de Jesus Antonio Sanchez Ramos qui a démarré en janvier 2023.

9.1.2 Développer la prise en compte de la sécurité dans l'ingénierie orientée mission

Avec l'accroissement toujours plus grand de la menace Cyber, les méthodes utilisées pour développer les SoS devront de plus en plus investir le champ de la cybersécurité. Dans les travaux précédents, nous avons introduit le paradigme "mission" pour permettre de disposer d'un invariant guidant l'évolution du SoS dans le temps. Il est clair que désormais, la mission d'un SoS ne devra plus être seulement réalisée, mais l'accomplissement de la mission devra l'être de manière sûre, au regard de la sécurité. Cela pose plusieurs enjeux. D'une part, comment spécifier les critères de sécurité à prendre en compte dans la mission, au delà du triptyque habituel *Disponibilité*, *Intégrité* et *Confidentialité* ? Comment imaginer l'évolution de la sécurité au fur-et-à-mesure du temps ? Comment s'assurer du maintien du niveau de sécurité tout au long de la mission ? Les questions sont nombreuses et relèvent à la fois de la capacité à imaginer la sécurité du SoS dans le temps, mais également de la capacité à maintenir le niveau exigé malgré l'évolution de ce dernier.

9.1.3 Vers une architecture soutenant une approche de sécurité centrée sur les données

Les systèmes actuels génèrent et gèrent de plus en plus de données, cela de manière partagées. Ces données transitent par des systèmes de systèmes de tailles et de structures différentes, et supportant des politiques de sécurités hétérogènes (cloud, capteurs, serveurs distants, ...). Cela engendre un risque supérieur d'accès malveillants à ces données et cela pour toutes les propriétés de sécurité (Intégrité, Disponibilité, Confidentialité) d'autant que l'aspect protéiforme des supports d'échange et de traitement des données, par sa complexité, augmente la taille de la surface d'attaque. Dans ce contexte, le besoin de disposer d'une sécurité centrée sur les données est fort.

La sécurité centrée sur les données ne change pas les fondamentaux de la sécurité traditionnelle. Il faut la voir comme un nouvel élément de défense en profondeur venant en complément de la défense périmétrique traditionnelle. Plusieurs démarches visent à traiter ce problème comme par exemple l'architecture *Zero Trust* [Rose et al., 2020]. Cependant l'ANSSI, si elle reconnait la pertinence de cette direction, met en garde contre la faible maturité des solutions présentées actuellement [ANSSI, 2021].

Mettre en place une sécurité centrée sur les données requiert plusieurs axes d'effort. Dans un premier temps, il faut déterminer une politique de contrôle d'accès à la fois modulable, évolutive et fiable. Pour cela, nous proposons de nous inspirer de l'IGI 1300 [SGDSN, 2021]

qui décrit les règles pour assurer la protection du secret de la Défense Nationale. Ces règles reposent sur trois concepts :

— Une information protégée se voit attribuer un niveau de *classification*, au plus juste, pour concilier la circulation de l'information nécessaire à l'efficacité opérationnelle tout en empêchant une divulgation excessive, potentiellement nuisible, qui donnerait à l'adversaire l'opportunité d'empêcher l'opération.
— Pour accéder à une information, une personne doit être *habilitée* au niveau de classification de l'information, attestant la confiance accordée à cette personne.
— Le *besoin d'en connaître* conceptualise par ailleurs que l'accès à une information classifiée est motivé par l'exercice de la fonction ou l'accomplissement d'une mission. En complément de l'habilitation, le besoin d'en connaître contextualise donc les demandes au regard du besoin opérationnel pour décider d'accorder ou non l'accès.

Se pose alors la problématique d'automatiser la manière de prendre ainsi en compte ce contexte opérationnel lors des décisions d'accorder ou non l'accès. Pour cela, nous proposons de définir et développer une architecture supportant les mécanismes nécessaire à l'implémentation d'un tel contrôle d'accès. Cela passe notamment par la capacité de développer des outils de chiffrement capables de supporter la politique d'accès. Pour cela, nous envisageons de nous inspirer des approches telles que le chiffrement basé sur les attributs (*Attribute-Based Encryption*, ABE [Sahai and Waters, 2005]) qui est une forme de chiffrement asymétrique dont le déchiffrement est conditionné par la nature du profil du destinataire, à savoir le fait qu'il possède - ou pas - un certain nombre de caractéristiques - les attributs.

Nous avons posé les bases de cette problématique dans un article introductif [Alquié et al., 2022] où nous décrivons d'une part le type de scénarios opérationnels que nous envisageons à travers des exemples et d'autre part une introduction à la manière dont un tel contrôle d'accès pourrait s'appuyer sur des mécanismes existants pour soutenir les travaux que nous projetons sur ce sujet.

Ce sujet de recherche a reçu le soutien de Thales à travers le financement d'une chaire de recherche en partenariat avec l'AMSCC et la Fondation St-Cyr qui a débuté ses travaux en janvier 2023. Elle dispose d'un financement assuré de 4 années comprenant le financement de plusieurs thèses. Je suis le titulaire de cette chaire. Nous venons de lancer le premier doctorant sur le sujet le 9 octobre 2023 : Etienne Lemonnier.

9.2 L'humain, un système comme les autres ?

Lors de la conférence ECSA'22, David Garlan a donné une conférence invitée intitulée "Humanizing Software Architecture". Lors de cette présentation il a notamment dressé le constat que les humains étaient généralement exclus des conceptions architecturales. Or, d'après lui, il y a des avantages à les intégrer en tant qu'entités de première classe de manière à améliorer la synergie homme-système requises dans les systèmes actuels. Cette vision rejoint celle défendue par les travaux de thèse de Paul Perrotin [Perrotin, 2022] dans lesquels des architectures composées d'humains étaient spécifiées de manière à en déterminer les risques de vulnérabilité humaine. Dans ce contexte, je souhaite contribuer à cette vision en développant les deux axes de recherche suivants.

La vision que nous avons portée dans [Perrotin, 2022] était qu'un humain peut-être considéré comme un système comme les autres, d'un point de vue architectural. En cela, le concept de système de systèmes peut alors englober le concept de système de systèmes socio-techniques en mettant au même niveau systèmes techniques et humains. L'objectif est ainsi de favoriser une meilleure synergie afin d'aboutir à de meilleures conceptions et de

réduire le risque de mauvaise prise en compte des aspects humains et/ou d'une mauvaise interaction entre les deux types de systèmes.

D'un point de vue conception architecturale, il est pertinent d'étendre les propriétés humaines prises en compte pour aller au-delà de la seule prise en compte de la vulnérabilité. L'idée n'est encore pas d'être exhaustif dans la représentation humaine, la complexité humaine rend cela impossible, mais de proposer un ensemble de propriétés humaines couvrant diverses caractéristiques que cherche à étudier et/ou spécifier l'architecte, cela selon plusieurs critères de qualité. Ces critères de qualité peuvent être les critères traditionnels tels qu'identifiés par [Barbacci et al., 1995] ou des critères plus spécifiques aux systèmes de systèmes comme ceux résumés par [Bianchi et al., 2015]. Il y a là une première question de recherche à trancher. Quels critères de qualités sont pertinent pour proposer une architecture dans laquelle certains sous-systèmes sont humains ? Quels sont les impacts en termes d'architecture sur cette approche et quelles en sont les conséquences sur les outils et méthodes existants ? Les questions sont nombreuses mais les enjeux importants.

D'autre part, dans [Perrotin, 2022], nous avons considéré une architecture composée uniquement d'humains, vus comme des systèmes en relation. Nous avons conçu une manière de caractériser l'humain à travers des propriétés pouvant avoir un impact sur sa vulnérabilité, un langage de modélisation permettant de représenter des architectures socio-techniques composées d'humains, et une méthode outillée permettant d'estimer la vulnérabilité d'un humain sous le coup d'une cyber-attaque, ainsi que la probabilité de propagation de cette vulnérabilité au reste de l'architecture modélisée. Dans la continuité de ces travaux, il est pertinent d'étudier la manière dont une vulnérabilité humaine ainsi détectée pourrait se propager aux systèmes physiques avec lesquels l'humain vulnérable est en contact d'une part, et d'autre part d'étudier la manière dont une vulnérabilité humaine pourrait se combiner à une vulnérabilité technique. L'idée principale ici soutenue est que la prise en compte simultanée de la vulnérabilité humaine et de la vulnérabilité technique des systèmes considérés peut aider à déterminer l'émergence de nouvelles formes de vulnérabilités lorsque ces deux types de systèmes sont mis en relation. D'autre part, représenter de telles architectures, dans lesquels opérateurs humains et systèmes techniques interopèrent, permet leur confrontation à des mécanismes telles que les arbres d'attaque complexes. En effet, il est notable que de nombreuses cyber-attaques sont construites en combinant à la fois des attaques ciblant les humains (désinformation, stress . . .) et des attaques plus techniques. Pour cela, il conviendra donc de développer des recherches d'une part sur le lien humain-système technique, de le caractériser, d'étudier sa porosité et les conséquences de cette porosité. D'autre part, un lien devra être fait avec les techniques d'attaques complexes telles que celles identifiées dans la matrice d'attaque MITRE ATT&CK [MITRE, 2023] ou dans la plateforme DISARM [DISARM, 2023].

Ce sujet de recherche a reçu le soutien de la DGA-MI. Cette dernière met à notre disposition Paul Perrotin, qui y est désormais chercheur, à hauteur de 50 % de son temps de travail.

9.3 Perspectives spécifiques à la Défense

Dans la suite directe de mon activité de recherche, certains projets que je souhaite développer sont plus spécifiques au domaine applicatif de la Défense. Je présente ici deux projets de recherche qui sont dans ce contexte.

9.3.1 Vers un IDM4Mili

Avec le déploiement du programme Scorpion au sein des forces terrestres françaises, le champ de bataille du futur entre de plein pied dans le domaine du traitement massif des données. Cela ouvre des possibilités intéressantes pour associer l'outil numérique au sens large aux actions de combat afin de développer une supériorité capacitaire. En effet, disposer des données en temps réel permet de les traiter de manière à aider à la prise de décision. Ainsi, la compréhension du déroulé d'une opération, là où elle était essentiellement humaine, peut-être associée à des outils numériques. Plusieurs applications peuvent en découler : détection de signaux faibles, détection de patrons d'attaques et/ou de défense à l'image de ceux décrits par Yakovleff [Yakovleff, 2006], utilisation d'intelligences artificielles pour aider la décision du commandement humain, etc. Pour cela cependant, il convient d'être en capacité de décrire les attendus de l'opération en cours. Nous avons dans des travaux précédents proposé d'utiliser l'ingénierie dirigée par les modèles (IDM) comme pierre angulaire d'un ou de plusieurs systèmes supportant de telles fonctionnalités. Nous avons développé des preuves de concepts autour d'un langage d'ordres utilisable à la fois pour la description des ordres d'opération, pour une vision commune du théâtre d'opérations au niveau des centre d'opérations (CO) mais également à un niveau micro pour le commandement de sections hybrides hommes-robots.

Plusieurs axes de recherche pourraient être menés à partir de nos travaux initiaux. En premier lieu, nous avons surtout envisagé le concept d'ordre à travers une vue structurelle. Cette démarche est assez naturelle en conception car la vue statique est toujours plus aisée à appréhender. Je propose donc d'étendre les travaux que nous avons menés sur le développement d'un langage d'ordres en prenant en compte une représentation dynamique de ces derniers. Pour cela il conviendra de développer des artéfacts à même de supporter la description comportementale associée à un ordre par exemple. L'idée est d'être capable de décrire le mouvement attendu d'une unité à travers des séquences de fragments comportementaux. Ainsi, il pourrait être possible de suivre son avancée, de voir si l'avancée prévue est en adéquation avec l'avancée réalisée, si le comportement attendu correspond aux patrons connus et spécifiés, ... Dans cette optique, il est possible de s'inspirer de travaux tels que ceux menés par Hanh Nhi Tran [Nguyen et al., 2022] qui vient juste de rejoindre notre équipe à St-Cyr.

Un deuxième axe de recherche que je souhaite développer vise à intégrer des éléments de prise en compte des moyens propres à la guerre hybride dans la conception des ordres d'opérations. En effet, comme le montre la guerre russo-ukrainienne actuelle, les opérations deviennent grandement hybrides : sont associées des effets cybers (et de tous niveaux, à la fois techniques, informationnels ...) aux effets cinétiques. Ainsi, je pense intéressant d'étendre les travaux menés autour du langage d'ordre d'opérations de manière à prendre en compte cette diversité. Dans les travaux précédents, nous nous sommes concentrés sur une seule famille d'ordre, spécifiée par la norme PROTERRE [Armée de Terre Française, 2014]. Il s'agira là d'étendre le domaine de couverture des ordres pris en charge par le langage, mais également d'y intégrer les ordres propres aux notions de cyber-attaques et cyber-défenses et de champs immatériels.

Enfin, du point de vue du programme Scorpion, le système SICS en est le système de commandement. Il a été développé par une entreprise de la BITD. Un axe de travail intéressant pourrait être d'étudier la faisabilité d'une modification d'un tel système en le construisant au-dessus du langage d'ordre que nous proposons de développer. Outre le fait que cela le rendrait plus modulaire et plus évolutif, du seul point de vue de la recherche, cela propose des perspectives intéressantes en termes de capacité à déployer des concepts IDM sur un système opérationnel, voire d'en faire la clé de voute de l'interopérabilité entre

systèmes de combat.

9.3.2 Développer des méthodes et des outils défensifs dans le cadre de la LII

A l'intersection des problématiques de cybersécurité et des problématiques de traitement de la donnée, la guerre informationnelle fait rage. La France, longtemps en retrait sur ce sujet, en subit des conséquences directe, que cela soit dans le domaine de la contre ingérence économique et/ou politique. On peut citer comme exemple le schéma désormais maintes fois répété en Afrique (Centrafrique, Mali et Burkina Faso) à travers lequel des campagnes de désinformation conduisent à un rejet de l'influence française au profit d'une ingérence russe et au déploiement des mercenaires de Wagner. On peut également citer l'affaire des sous-marins australiens au détriment de Naval Group. En décembre 2022, le président Macron a, en réaction à cette situation, décidé que la lutte informatique d'influence devenait une des fonctions stratégiques de la France.

Dans l'approche qui est la mienne et qui considère l'humain comme un système à part entière, les opérations d'influence sont vues comme des cyber-attaques au même titre que les attaques techniques le sont pour les systèmes techniques. Dans ce cas, avant même de pouvoir mettre en œuvre des contremesures permettant de lutter contre ces attaques, il faut être capable de les détecter. Or, cette détection peut-être complexe puisque les attaques sont souvent menées sur de longues périodes (plusieurs mois à plusieurs années) et prennent souvent des formes complexes. Le parallèle avec la manière dont se déroulent des cyber-attaques physiques est adéquat.

Dans ce contexte, je propose un axe de recherche spécifique à la LII mais s'appuyant sur cette vue de l'humain en tant que système et dans lequel je cherche à mobiliser mon expérience en ingénierie au sens large pour développer des outils et méthodes visant à réduire les vulnérabilités cyber de niveau informationnel auxquelles pourraient être soumis des systèmes humains. Ces travaux sont typiquement interdisciplinaires. L'expertise que j'amène doit forcément s'appuyer sur une connaissance fine du système humain. Cette dernière est apportée par différents domaines issus des sciences humaines et sociales (sociologie, psychologie, marketing ...). Mon rôle est alors de transposer cette connaissance sous la forme de modèles ouvrant la porte à un certain nombre de traitements.

Ces travaux sont déjà commencés dans le cadre d'une convention entre le ministère des Armées et l'AMSCC, dans laquelle je suis impliqué et collabore avec une équipe de SHS de l'AMSCC et le ministère. Il est prévu d'étendre cette convention dans les mois à venir de manière à financer un projet.

9.3.2.1 Sensibiliser aux vulnérabilités liées à l'emprunte numérique

Le premier axe de recherche que je souhaite développer vise à réduire la surface d'attaque d'un humain. En effet, plus l'attaquant possède d'information sur un système, plus il est facile pour lui de mobiliser des attaques ciblées et de maximiser ainsi les chances de réussite de ces dernières. Ainsi, les traces numériques laissées, volontairement ou non, par un humain sur la Toile sont autant d'informations exploitables de manière malveillante. Il y a donc un intérêt majeur à réduire ces traces que l'on appelle emprunte numérique. Dans ce contexte, je propose de développer une méthode visant à détecter dans un premier temps l'emprunte numérique d'un humain, notamment dans un objectif de sensibilisation. En effet, autant toute personne est libre d'utiliser la Toile comme elle l'entend dans la limite de ce que la loi autorise, autant comprendre les enjeux liées aux traces que cela génère est une nécessité.

Comme pour la thématique précédente, ces travaux sont en déjà commencés et se déroulent dans le même contexte collaboratif.

9.3.2.2 Vers une explicabilité des Fake News basée modèle

Le deuxième axe de recherche que je souhaite développer sur ce sujet s'appuie sur les résultats des travaux que j'ai conduits en amont sur la modélisation du concept de Fake News. Nous avons proposé précédemment un modèle conceptuel caractérisant le concept de Fake News [Belloir et al., 2022b]. Ce modèle identifie différents niveaux sémantiques dans la définition d'une Fake News : un niveau syntaxique à travers lequel on peut caractériser la structure de la Fake News, un niveau sémantique qui décrit le fonctionnement de la Fake News sur sa cible et un niveau opérationnel qui décrit le contexte et l'objectif opérationnel d'une Fake News. Un problème sous-jacent de la désinformation liée au Fake News est de convaincre un auditoire qui a été sensible au message d'une Fake News que ce dernier est faux, ou a minima orienté. En effet, comme le soulignent [Zhou and Zafarani, 2020], malgré le nombre croissant de travaux autour du concept de Fake News, la façon dont on peut évaluer automatiquement l'authenticité des nouvelles d'une manière efficace et explicable reste une question ouverte. Pour cela, il ne suffit pas de dire si une information est ou n'est pas une Fake News. Ainsi, les nombreux outils, qui font de la détection de Fake News à partir d'outils construits sur du Machine Learning (ML) par exemple, ne sont pas suffisants. Ils donnent un résultat sans le justifier. Le modèle que nous avons proposé dans [Belloir et al., 2022b] s'inscrit dans une démarche visant à produire des modèles compréhensibles, dignes de confiance et gérables par les humains, comme le suggère [Gunning et al., 2019]. En effet, faciliter l'explicabilité et l'interprétabilité a suscité beaucoup d'intérêt dans la recherche en intelligence artificielle dans le but de produire des systèmes intelligents qui renforcent la confiance des utilisateurs par la compréhension des raisonnements et des automatismes sous-jacents [Gunning et al., 2019]. Pour le contexte des Fake News, à notre connaissance, les travaux impliquant la fonction d'explicabilité dans les méthodes de détection des Fake News en sont encore à leurs débuts [Shu et al., 2019].

Dans ce contexte, je propose d'utiliser le modèle conceptuel comme pierre angulaire d'une approche d'explicabilité. Pour cela, j'envisage une approche basée sur l'utilisation d'outils de ML pour identifier dans une information les caractéristiques typiques d'une Fake News, décrites par le modèle conceptuel. Ainsi, si une nouvelle "matche" avec le modèle conceptuel, on peut expliquer par le modèle conceptuel que c'est bien une Fake News. Par ailleurs, souvent les campagnes de désinformation sont protéiformes. Récemment, on a montré qu'un certain nombre de Fake News à propos de la pandémie circulaient dans les même cercles, voire étaient issues des mêmes comptes que des Fake News pro-russes traitant du conflit en Ukraine. Faire un lien entre ces fausses nouvelles permettrait de montrer l'origine de certaines désinformations de manière claire. Pour cela encore, le modèle à travers sa caractérisation de l'objectif opérationnel peut être un outil convainquant. Je propose donc de coupler son utilisation avec des outils de détections de cyber-attaques tels que la plateforme Disarm [DISARM, 2023]. Cela permettra d'identifier des arbres d'attaque contre lesquels il sera alors possible de construire des contremesures.

Ce sujet de recherche a reçu le soutien de l'Université de Bretagne Sud à travers le financement d'une bourse de doctorat qui a débuté en décembre 2022 pour 3 ans.

9.4 Remarques

A ce jour, la moitié des perspectives identifiées et présentées en amont ont déjà un financement allant de 2 à 4 ans. Une vue synthétique des sujets et des financements est

donnée par la Table 9.1.

Perspective	Sous-Perspective	Financement	Durée	Période
Ingénierie pour la sécurité	Renforcer le lien entre experts en sécurité et architectes	Thèse de de Jesus Antonio Sanchez Ramos - Financement Pôle d'Excellence Cyber (DGA)	3 ans	01/2023 12/2026
	Security for Mission Engineering	*Non encore financé*	?	
	Data Centric Security	Chaire " Cyberdéfense - Cybersécurité " St Cyr Thales	4 ans	01/2023 12/2027
L'humain un système comme un autre	Lien opérateur humain - Système	chercheur DGA à mi-temps	2 ans (renouvelable)	09/2023 08/2025
Applicatifs militaires	IDM4Mili	*Non encore financé*	?	
	LII et guerre informationnelle	Thèse Angélique Yameogo (financement UBS)	3 ans	12/2022 11/2025

TABLE 9.1 – Vue synthétique des axes de recherche futurs et de leurs financements

Bibliographie

[Ahamd, 2013] Ahamd, M. (2013). *Modélisation et vérification des exigences fonctionnelles et non fonctionnelles des systèmes ambiants, auto adaptatifs*. PhD thesis, Université Toulouse 2 Le Mirail. (Cité en pages vii, 16 et 18.)

[Ahmad et al., 2013a] Ahmad, M., Araújo, J., Belloir, N., Bruel, J.-M., Gnaho, C., Laleau, R., and Semmak, F. (2013a). Self-Adaptive Systems Requirements Modelling : four related approaches comparison. In *Proceedings of the Comparing Requirements Modeling Approaches (CMA@RE'13) workshop, in the field of IEEE International Conference on Requirements Engineering 2013*, pages 37–42, Rio de Janeiro, Brasil. IEEE Computer Press. (Cité en pages 13 et 15.)

[Ahmad et al., 2015] Ahmad, M., Belloir, N., and Bruel, J.-M. (2015). Modeling and verification of Functional and Non-Functional Requirements of ambient Self-Adaptive Systems. *Journal of Systems and Software*, 107(C) :50–70. (Cité en pages 13, 17 et 88.)

[Ahmad et al., 2013b] Ahmad, M., Dragomir, I., Bruel, J.-M., Ober, I., and Belloir, N. (2013b). Early Analysis of Ambient Systems SYSML Properties using OMEGA2-IFx. In *Proceedings of the 3rd International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH'13)*, pages 147–154, Reykjavik, Iceland. SciTePress. (Cité en pages 13 et 17.)

[Alquié et al., 2022] Alquié, D., Belloir, N., Buisson, J., and Touseau, L. (2022). Vers la sécurité dans un environnement opérationnel collaboratif dynamique. In *Proceedings of the 29th Computer & Electronics Security Application Rendezvous co-located with the 7th European Cyber Week (ECW 2022)*. (Cité en page 70.)

[ANSSI, 2021] ANSSI (2021). Avis scientifique et technique : le modèle zero trust. Technical report, Agence nationale de la sécurité des systèmes d'information. https://www.ssi.gouv.fr/agence/publication/le-modele-zero-trust/. (Cité en page 69.)

[ANSSI, 2023] ANSSI (2023). Panorama de la cyberenace 2022. Technical report, Agence nationale de la sécurité des systèmes d'information (ANSSI). (Cité en page 68.)

[Armée de Terre Française, 2014] Armée de Terre Française (2014). TTA 150 : Le combat PROTERRE en milieu ouvert. Manuel de combat, Armée de Terre Française. (Cité en pages 48 et 72.)

[Barbacci et al., 1995] Barbacci, M., Klein, M., Longstaff, T., and Weinstock, C. (1995). Quality attributes. Technical Report CMU/SEI-95-TR-021, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA. http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=12433. (Cité en page 71.)

[Belloir et al., 2014] Belloir, N., Bruel, J.-M., and Faudou, R. (2014). Modélisation des exigences en UML/SysML. *revue Génie Logiciel. Numéro spécial Ingénierie des Exigences*, (111) :6–12. (Cité en pages 13 et 15.)

[Belloir et al., 2008] Belloir, N., Bruel, J.-M., Hoang, N., and Pham, C.-D. (2008). Utilisation de SysML pour la modélisation des réseaux de capteurs. In *Actes de la conférence Langages et Modèles à Objets (LMO'08)*, pages 171–186, Montreal, Canada. RNTI. (Cité en page 15.)

[Belloir et al., 2019] Belloir, N., Buisson, J., and Bartheye, O. (2019). Metamodeling NATO Operation Orders : a proof-of-concept to deal with digitalization of the battlefield. In *Proceedings 14th IEEE System of Systems Engineering Conference*, pages 260–265. IEEE. (Cité en pages 45, 47 et 174.)

[Belloir et al., 2022a] Belloir, N., Buisson, J., and Touseau, L. (2022a). Model-driven engineering as the interface for tactical operation order of mixed robot/human platoons. In Rocha, Á., Fajardo-Toro, C. H., and Rodríguez, J. M. R., editors, *Developments and Advances in Defense and Security, proceedings of the 2021 Multidisciplinary International Conference of Research Applied to Defense and Security*, pages 205–214, Singapore. Springer Singapore. (Cité en pages 45, 48 et 88.)

[Belloir et al., 2022b] Belloir, N., Ouerdane, W., and Pastor, O. (2022b). Characterizing fake news : A conceptual modeling-based approach. In *2022 41th International Conference on Conceptual Modeling (ER22)*, pages 115–129. (Cité en pages 51, 52, 54, 74 et 88.)

[Belloir et al., 2022c] Belloir, N., Ouerdane, W., Pastor, O., Frugier, É., and de Barmon, L.-A. (2022c). A conceptual characterization of fake news : A positioning paper. In Guizzardi, R., Ralyté, J., and Franch, X., editors, *Research Challenges in Information Science*, pages 662–669, Cham. Springer International Publishing. (Cité en pages vii, 51, 52, 53 et 54.)

[Bianchi et al., 2015] Bianchi, T., Santos, D. S., and Felizardo, K. R. (2015). Quality attributes of systems-of-systems : A systematic literature review. In *2015 IEEE/ACM 3rd International Workshop on Software Engineering for Systems-of-Systems*, pages 23–30. (Cité en page 71.)

[Brereton et al., 2008] Brereton, P., Kitchenham, B., Budgen, D., and Li, Z. (2008). Using a protocol template for case study planning. In *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, EASE'08, page 41–48, Swindon, GBR. BCS Learning & Development Ltd. (Cité en page 40.)

[Bruel et al., 2009] Bruel, J.-M., Belloir, N., and Manzoor, A. (2009). SPAS : un profil SysML pour les systèmes auto-adaptatifs. In *15ème Colloque National de la Recherche en IUT (CNRIUT), Lille, 08/06/09-10/06/09*. (Cité en page 15.)

[Buisson et al., 2020] Buisson, J., Belloir, N., and Mbeck, J. L. (2020). Digitalization in Next Generation C2 : Research Agenda from Model-Based Engineering Perspective. In *Proceedings 15th IEEE System of Systems Engineering Conference*, pages 243–248. IEEE. (Cité en pages 45 et 48.)

[CCDH, 2021] CCDH (2021). The disinformation dozen - why platforms must act on twelve leading online anti-vaxxers. Technical report, Center for Countering Digital Hate. `https://www.counterhate.com/disinformationdozen`. (Cité en page 53.)

[Centre de doctrine d'emploi des forces (CDEF), 2014] Centre de doctrine d'emploi des forces (CDEF) (2014). Méthodologie d'élaboration d'une décision opérationnelle tactique. Technical Report CDT 60.001, Armée de terre. (Cité en page 46.)

[Cherfa, 2022] Cherfa, I. (2022). *Mission Oriented Process for Systems of Systems Engineering*. PhD thesis, Université de Blisa 1. (Cité en pages vii, 19, 20, 21 et 22.)

[Cherfa et al., 2019] Cherfa, I., Belloir, N., Sadou, S., Fleurquin, R., and Bennouar, D. (2019). Systems of systems : From mission definition to architecture description. *Systems Engineering*, 22(6) :437–454. (Cité en pages 19, 20, 21 et 88.)

[Cherfa et al., 2018] Cherfa, I., Sadou, S., Belloir, N., and Fleurquin, R. (2018). Involving the Application Domain Expert in the Construction of Systems of Systems. In *Proceedings of the 13th System of Systems Engineering Conference (SoSE'18)*, pages 335–342. IEEE. (Cité en pages 19 et 20.)

[Dalpiaz et al., 2016] Dalpiaz, F., Paja, E., and Giorgini, P. (2016). *Security Requirements Engineering : Designing Secure Socio-Technical Systems*. MIT Press. (Cité en page 37.)

[DISARM, 2023] DISARM (acceded 01-02-2023). Disarm framework. https://www.disarm.foundation/framework. (Cité en pages 71 et 74.)

[Gnaho and Semmak, 2011] Gnaho, C. and Semmak, F. (2011). Une extension sysml pour l'ingénierie des exigences non fonctionnelles orientée but. *Ingénierie des Systèmes d Information*, 16(1) :9–32. (Cité en page 16.)

[Gorod et al., 2014] Gorod, A., White, B. E., Ireland, V., Gandhi, S. J., and Sauser, B. (2014). *Case Studies in System of Systems, Enterprise Systems, and Complex Systems Engineering*. CRC Press. (Cité en page 23.)

[Goya, 2014] Goya, M. (2014). *Sous le feu, la mort comme hypothèse de travail*. Broché. (Cité en page 46.)

[Gunning et al., 2019] Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., and Yang, G.-Z. (2019). XAI : Explainable Artificial Intelligence. *Science Robotics*, 4(37) :eaay7120. (Cité en page 74.)

[Hachem et al., 2020] Hachem, J. E., Chiprianov, V., Babar, M. A., Khalil, T. A., and Aniorte, P. (2020). Modeling, analyzing and predicting security cascading attacks in smart buildings systems-of-systems. *Journal of Systems and Software*, 162 :110484. (Cité en page 68.)

[Kopetz et al., 2015] Kopetz, H., Höftberger, O., Frömel, B., Brancati, F., and Bondavalli, A. (2015). Towards an understanding of emergence in systems-of-systems. In *2015 10th System of Systems Engineering Conference (SoSE)*, pages 214–219. (Cité en page 68.)

[Lutz, 1993] Lutz, R. R. (1993). Targeting safety-related errors during software requirements analysis. In *Proceedings of the 1st ACM SIGSOFT Symposium on Foundations of Software Engineering*, SIGSOFT '93, page 99–106, New York, NY, USA. Association for Computing Machinery. (Cité en page 14.)

[Mankins, 1995] Mankins, J. (1995). Technology readiness level – a white paper. (Cité en page 49.)

[Messe, 2021] Messe, N. (2021). *Security by Design : An asset-based approach to bridge the gap between architects and security experts*. PhD thesis, Université de Bretagne Sud. (Cité en pages vii, 31, 32 et 34.)

[Messe et al., 2020a] Messe, N., Chiprianov, V., Belloir, N., El Hachem, J., Fleurquin, R., and Sadou, S. (2020a). Asset-Oriented Threat Modeling. In *The 19th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (IEEE TrustCom 2020)*. IEEE. (Cité en pages 29, 33, 68 et 88.)

[Messe et al., 2019] Messe, N., **Belloir, Nicolas**, Chiprianov, V., Cherfa, I., Fleurquin, R., and Sadou, S. (2019). Development of Secure Systems of Systems Needing a Rapid Development. In *Proceedings 14th IEEE System of Systems Engineering Conference*. IEEE. (Cité en page 29.)

[Messe et al., 2020b] Messe, N., **Belloir, Nicolas**, Chiprianov, V., El Hachem, J., Fleurquin, R., and Sadou, S. (2020b). An Asset-Based Assistance for Secure by Design. In *The 27th Asia-Pacific Software Engineering Conference (APSEC 2020)*. IEEE-CS. (Cité en pages vii, 29, 31, 33 et 68.)

[Microsoft Corporation, 2018] Microsoft Corporation (2018). SDL Threat Modeling Tool. Security Development Lifecycle. https://www.microsoft.com/en-us/sdl/adopt/threatmodeling.aspx. (Cité en page 33.)

[MITRE, 2023] MITRE (acceded 01-02-2023). Mitre att&ck. https://attack.mitre.org/. (Cité en page 71.)

[Naouar et al., 2021] Naouar, D., Hachem, J. E., Voirin, J.-L., Foisil, J., and Kermarrec, Y. (2021). Towards the integration of cybersecurity risk assessment into model-based requirements engineering. In *2021 IEEE 29th International Requirements Engineering Conference (RE)*, pages 334–344. (Cité en page 68.)

[NATO, 2000] NATO (2000). STANAG 2014 : Formats for Orders and Designation of Timings, Locations and Boundaries. Technical Report MAS(ARMY)0307-TOP/2014, NATO Military Agency for Standardization. (Cité en pages 46 et 47.)

[Nguyen et al., 2022] Nguyen, M. K., Tran, H. N., and Ober, I. (2022). Process Mining to Discover the Global Process from its Fragments' Executions. In *17th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2022)*, pages 363–370, Online Streaming, France. SCITEPRESS - Science and Technology Publications. (Cité en page 72.)

[Ober and Dragomir, 2010] Ober, I. and Dragomir, I. (2010). Omega2 : A new version of the profile and the tools. In *2010 15th IEEE International Conference on Engineering of Complex Computer Systems*, pages 373–378. (Cité en page 17.)

[Object Management Group, 2017] Object Management Group (2017). OMG® Unified Modeling Language® (OMG UML®) - Version 2.5.1. Specification Document formal/2017-12-05, OMG. (Cité en page 14.)

[Object Management Group, 2019] Object Management Group (2019). OMG Systems Modeling Language (OMG SysML™) - Version 1.6. Specification Document formal/19-11-01, OMG. (Cité en page 14.)

[(OMG), 1997] (OMG), O. M. G. (1997). Unified modeling language (UML) version 1.0. Standard, Object Management Group (OMG). (Cité en page 2.)

[Perrotin, 2022] Perrotin, P. (2022). *Analyse de la vulnérabilité humaine dans les systèmes de systèmes socio-techniques*. PhD thesis, Ecole Nationale Supérieure Mintes-Télécom Atlantique Bretagne Pays de la Loire. (Cité en pages 35, 37, 38, 40, 70 et 71.)

[Perrotin et al., 2022a] Perrotin, P., Belloir, N., Sadou, S., Hairion, D., and Beugnard, A. (2022a). Hos-ML : Socio-Technical System ADL Dedicated to Human Vulnerability Identification. In *26th International Conference on Engineering of Complex Computer Systems (ICECCS)*, pages 1–6, Hiroshima City, Japan. (Cité en page 35.)

[Perrotin et al., 2022b] Perrotin, P., Belloir, N., Sadou, S., Hairion, D., and Beugnard, A. (2022b). Using the architecture of Socio-Technical System to analyse its vulnerability. In *17th Annual System of Systems Engineering Conference (SOSE)*, pages 361–366. (Cité en page 35.)

[Perrotin et al., 2022c] Perrotin, P., Belloir, N., Sadou, S., Hairion, D., and Beugnard, A. (2022c). Using the architecture of socio-technical system to analyse its vulnerability. In *2022 17th Annual System of Systems Engineering Conference (SOSE)*, pages 361–366. (Cité en page 88.)

[Radio Technical Commission for Aeronautics, 1992] Radio Technical Commission for Aeronautics (1992). RTCA/DO-178B, Software Considerations in Airborne Systems and Equipment Certification,. Standard, Radio Technical Commission for Aeronautics. (Cité en page 2.)

[Rose et al., 2020] Rose, S., Borchert, O., Mitchell, S., and Connelly, S. (2020). Zero trust architecture. Technical report, Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD. `https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=930420`. (Cité en page 69.)

[Sahai and Waters, 2005] Sahai, A. and Waters, B. (2005). Fuzzy Identity-Based Encryption. In *Advances in Cryptology – EUROCRYPT 2005*, Lecture Notes in Computer Science, pages 457–473. (Cité en page 70.)

[SGDSN, 2021] SGDSN (2021). Instruction générale interministérielle n° 1300/SGDSN/PSE/PSD du 9 août 2021 sur la protection du secret de la défense nationale. https://www.ssi.gouv.fr/administration/reglementation/protection-des-systemes-informations/instruction-generale-interministerielle-n-1300-sur-la-protection-du-secret-de-la-defense-n (Cité en page 69.)

[Shevchenko et al., 2018] Shevchenko, N., Chick, T. A., O'riordan, P., Scanlon, T. P., and Woody, C. (2018). Threat modeling : a summary of available methods. *Software Engineering Institute. Carnegie Mellon University*. (Cité en page 33.)

[Shostack, 2008] Shostack, A. (2008). Experiences Threat Modeling at Microsoft. In Whittle, J., Jürjens, J., Nuseibeh, B., and Dobson, G., editors, *Proceedings of the Workshop on Modeling Security (MODSEC08) held as part of the 2008 International Conference on Model Driven Engineering Languages and Systems (MODELS) Toulouse, France, September 28, 2008*, volume 413 of *CEUR Workshop Proceedings*. CEUR-WS.org. (Cité en page 33.)

[Shu et al., 2019] Shu, K., Cui, L., Wang, S., Lee, D., and Liu, H. (2019). Defend : Explainable fake news detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 395–405, New York, NY, USA. Association for Computing Machinery. (Cité en page 74.)

[Sommerville and Sawyer, 1997] Sommerville, I. and Sawyer, P. (1997). *Requirements Engineering : A Good Practice Guide*. Wiley, New York, NY, USA, 1st edition. (Cité en page 14.)

[Sousa-Poza, 2015] Sousa-Poza, A. (2015). Mission engineering. *International Journal of System of Systems Engineering*, 6(3) :161–185. (Cité en page 20.)

[Spreeuwenberg, 2019] Spreeuwenberg, S. (2019). *AIX : Artificial Intelligence needs eXplanation : Why and how transparency increases the success of AI solutions*. LibRT BV, Amsterdam. (Cité en page 55.)

[Sun Tzu, v JC] Sun Tzu (IVe Av. JC). *L'art de la guerre*. (Cité en page 46.)

[Tøndel et al., 2010] Tøndel, I. A., Jensen, J., and Røstad, L. (2010). Combining misuse cases with attack trees and security activity models. In *2010 International Conference on Availability, Reliability and Security*, pages 438–445. (Cité en page 68.)

[van den Berghe et al., 2018] van den Berghe, A., Yskout, K., Scandariato, R., and Joosen, W. (2018). A lingua franca for security by design. In *2018 IEEE Cybersecurity Development (SecDev)*, pages 69–76. (Cité en page 27.)

[van Lamsweerde, 2009] van Lamsweerde, A. (2009). *Requirements Engineering : From system goals to UML models to software specifications*. John Wiley & Sons. (Cité en page 16.)

[van Lamsweerde and Letier, 2000] van Lamsweerde, A. and Letier, E. (2000). Handling obstacles in goal-oriented requirements engineering. *IEEE Transactions on Software Engineering*, 26(10) :978–1005. (Cité en page 15.)

[Vegas et al., 2015] Vegas, S., Apa, C., and Juristo, N. (2015). Cross-over designs in software engineering experiments : Benefits and perils. *IEEE Transactions on Software Engineering*, 42 :1–1. (Cité en page 34.)

[Verizon, 2016] Verizon (2016). Data Breach Investigations Report. Technical report, Verizon. https://www.verizon.com/about/news/verizons-2016-data-breach-investigations-report-finds-cybercriminals-are-exploiting-human, dernier accès le 30/08/2022. (Cité en page 36.)

[Von Clausewitz, 1832] Von Clausewitz, C. (1832). *De la guerre*. (Cité en page 46.)

[Wanderley et al., 2014] Wanderley, F., Belloir, N., Bruel, J.-M., Hameurlain, N., and Araújo, J. (2014). Des buts à la modélisation système : une approche de modélisation des exigences centrée utilisateur. In *Actes du XXXIIeme congrès INFormatique des Organisations et Systemes d'Information et de Decision (INFORSID)*, pages 113–128, Lyon, Inforsid. (Cité en pages 13 et 16.)

[Wanderley et al., 2012] Wanderley, F., da Silveira, D. S., Araujo, J. a., and Lencastre, M. (2012). Generating Feature Model from Creative Requirements Using Model Driven Design. In *16th Int. Software Product Line Conference - Volume 2*, SPLC'12, pages 18–25. ACM. (Cité en page 16.)

[Whittle et al., 2009] Whittle, J., Sawyer, P., Bencomo, N., Cheng, B. H., and Bruel, J.-M. (2009). Relax : Incorporating uncertainty into the specification of self-adaptive systems. In *2009 17th IEEE International Requirements Engineering Conference*, pages 79–88. (Cité en pages 15 et 16.)

[Yakovleff, 2006] Yakovleff, M. (2006). *Tactique théorique*. Broché. (Cité en pages 46 et 72.)

[Zeng et al., 2019] Zeng, J., Wu, S., Chen, Y., Zeng, R., Wu, C., and Caballero-Gil, P. (2019). Survey of attack graph analysis methods from the perspective of data and knowledge processing. *Sec. and Commun. Netw.*, 2019. (Cité en page 68.)

[Zhou and Zafarani, 2020] Zhou, X. and Zafarani, R. (2020). A survey of fake news : Fundamental theories, detection methods, and opportunities. *ACM Comput. Surv.*, 53(5). (Cité en page 74.)

Annexes

Annexe A

Modalités pour la rédaction du document d'HDR

Guillaume Gravier
Directeur de l'IRISA
guillaume.gravier@irisa.fr

À qui de droit.

Rennes, le 29 octobre 2021

Objet : modalités pour la rédaction du document d'habilitation à diriger des recherches

Chère collègue, cher collègue,

Vous êtes amené.e à évaluer un document d'habilitation à diriger des recherches d'un.e de nos collègues de l'IRISA et nous vous remercions d'avoir accepté ce travail.

Nous tenons par ce courrier à vous apporter quelques précisions sur les recommandations données aux candidat.e.s à l'HDR par le laboratoire.

Chaque candidat.e est bien évidemment libre de la forme et du contenu de son document d'HDR et le jury demeure le seul juge de la qualité de ce dernier et du mérite de la candidate ou du candidat. Toutefois, nous vous informons que l'IRISA recommande sur la forme un document d'HDR court (30 à 50 pages, hors articles joints et annexes) présentant une synthèse des travaux effectués et les perspectives qui en découlent. Le document peut être rédigé indifféremment en français ou en anglais.

En vous remerciant à nouveau pour le temps consacré à l'évaluation que vous menez, je vous prie de croire en l'assurance de ma cordiale considération.

Le directeur de l'IRISA
UMR 6074
Guillaume Gravier

Annexe B

Sélection d'articles scientifiques

Les articles suivants sont donnés afin de permettre au lecteur de juger de la qualité scientifique du travail de recherche présenté dans cette HDR. J'ai choisi de présenter un article pour illustrer chaque chapitre.

Chapitre	référence	sujet
2	[Ahmad et al., 2015] Manzoor Ahmad, Nicolas Belloir, and Jean-Michel Bruel. Modeling and Verification of Functional and Non-Functional Requirements of Ambient Self-Adaptive Systems. Journal of Systems and Software, 107(C) :50–70, sep 2015 (Core A)	Article de synthèse présentant les travaux réalisés pendant la thèse d'Ahmad Manzoor
3	[Cherfa et al., 2019] Imane Cherfa, Nicolas Belloir, Salah Sadou, Régis Fleurquin, and Djamal Bennouar. Systems of Systems : From Mission Definition to Architecture Description. Systems Engineering, 22(6) :437–454, 2019 (Q1/Q2 SCIMAGO)	Article de synthèse présentant les travaux réalisés pendant la thèse d'Imane Cherfa
4	[Messe et al., 2020a] Nan Messe, Vanea Chiprianov, Nicolas Belloir, Jamal El Hachem, Réis Fleurquin, and Salah Sadou. Asset-Oriented Threat Modeling. In Proceedings of the 19th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom'20), pages 491–501. IEEE, December 29 - January 1 2020 (Core A)	Article présentant la deuxième partie des travaux réalisés pendant la thèse de Nan Messe
5	[Perrotin et al., 2022c] Paul Perrotin, Nicolas Belloir, Salah Sadou, David Hairion, and Antoine Beugnard. Using the architecture of Socio-Technical System to analyse its vulnerability. In Proceedings of the 17th Annual System of Systems Engineering Conference (SOSE'22), pages 361–366, 2022 (conférence principale dans le domaine des SoS)	Article présentant de manière succincte les travaux menés pendant la thèse de Paul Perrotin
6	[Belloir et al., 2022a] Nicolas Belloir, Jérémy Buisson, and Lionel Touseau. Model-Driven Engineering as the Interface for Tactical Operation Order of Mixed Robot/Human Platoons. In Developments and Advances in Defense and Security, proceedings of the 2021 Multidisciplinary International Conference of Research Applied to Defense and Security, pages 205–214. Springer, 2022 (conférence spécialisé sur les problématiques de Défense. Publié sous forme de chapitre d'ouvrage chez Springer)	Dernier article sur notre approche appliquant l'IDMà la Défense. J'ai choisi de présenter celui-ci car il me semble révélateur du travail de recherche incluant des étudiants et se déroulant dans une école militaire
7	[Belloir et al., 2022b] Nicolas Belloir, Wassila Ouerdane, and Oscar Pastor. Characterizing Fake News : A Conceptual Modeling-based Approach. In Proceedings of the 41st International Conference on Conceptual Modeling (ER'22), pages 115–129, Cham, 2022. Lecture Notes in Computer Science, vol 13607. Springer (Core A)	Article présentant le modèle conceptuel des Fake News

The Journal of Systems and Software 107 (2015) 50–70

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Modeling and verification of Functional and Non-Functional Requirements of ambient Self-Adaptive Systems

Manzoor Ahmad [a], Nicolas Belloir [a,*], Jean-Michel Bruel [b]

[a] University of Pau and the Pays of the Adour, LIUPPA, 64000 Cedex, France
[b] University of Toulouse, CNRS/IRIT, F-31062 Toulouse Université Cedex, France

ARTICLE INFO

Article history:
Received 23 May 2014
Revised 14 May 2015
Accepted 15 May 2015
Available online 27 May 2015

Keywords:
Non Functional Requirements
Model Driven Engineering
Relax
Dynamic Adaptive Systems
Properties verification
Goal Oriented Requirements Engineering

ABSTRACT

Self-Adaptive Systems modify their behavior at run-time in response to changing environmental conditions. For these systems, Non-Functional Requirements play an important role, and one has to identify as early as possible the requirements that are adaptable. We propose an integrated approach for modeling and verifying the requirements of Self-Adaptive Systems using Model Driven Engineering techniques. For this, we use Relax, which is a Requirements Engineering language which introduces flexibility in Non-Functional Requirements. We then use the concepts of Goal-Oriented Requirements Engineering for eliciting and modeling the requirements of Self-Adaptive Systems. For properties verification, we use OMEGA2/IFx profile and toolset. We illustrate our proposed approach by applying it on an academic case study.

1. Introduction

As applications continue to grow in size, complexity, and heterogeneity, it becomes increasingly necessary for computing-based systems to dynamically self-adapt to changing environmental conditions. These systems are called Dynamically-Adaptive Systems (DASs) (Whittle et al., 2009). Example applications that require DASs capabilities include automotive systems, telecommunication systems, environmental monitoring, and power grid management systems. In this context, an adaptive system is a set of interacting or interdependent entities, real or abstract, forming an integrated whole that together are able to respond to environmental changes or changes in the interacting parts. Self Adaptive Systems (SAS) like other systems, have goals that must be satisfied and, whether these goals are explicitly identified or not, system requirements should be formulated to guarantee goal satisfaction. This fundamental principle has served systems development well for several decades but is founded on an assumption that goals are fixed. In general, goals can remain fixed if the environment in which the system operates is stable (Whittle et al., 2008). The distributed nature of SAS and changing environmental factors (including human interaction) makes it difficult to anticipate all the explicit states in which the system will be during its lifetime.

It is generally accepted that errors in requirements are very costly to fix (Lutz, 1993). The avoidance of erroneous requirements is particularly important for the emerging class of systems that need to adapt dynamically to changes in their environment. Many such DASs are being conceived for applications that require a high degree of assurance (Kasten et al., 2003), in which an erroneous requirement may result in a failure at run-time that has serious consequences. The requirement for high assurance is not unique to DASs, but the requirement for dynamic adaptation introduces complexity of a kind not seen in conventional systems where adaptation, if it is needed at all, can be done off-line. The consequent dynamic adaptation complexity is manifested at all levels, from the services offered by the run-time platform, to the analytical tools needed to understand the environment in which the DASs must operate.

Requirements Engineering (RE) is concerned with what a system ought to do and within which constraints it must do it. RE for SAS, therefore, must address what adaptations are possible and how those adaptations are carried out. In particular, questions to be addressed include: what aspects of the environment are relevant for adaptation? Which requirements are allowed to vary or evolve at run-time and which must always be maintained? In short, RE for SAS must deal with uncertainty because the expectations on the environment frequently vary over time. We identify the uncertainty in requirements of these systems and show how to verify it.

We are of the view that, on one hand, requirements for SAS should consider the notion of uncertainty while defining it; on the other hand, there should be a way to verify these requirements as early

* Corresponding author. Tel.: +33559407571; fax: +33559407654.
E-mail addresses: manzoor.ahmad@univ-pau.fr (M. Ahmad), nicolas.belloir@univ-pau.fr, nbelloir@gmail.com (N. Belloir), bruel@irit.fr (J.-M. Bruel).

http://dx.doi.org/10.1016/j.jss.2015.05.028
0164-1212/

as possible, even before the development of these systems starts. In order to handle the notion of uncertainty in SAS, RE languages for these systems should include explicit constructs for identifying the point of flexibility in its requirements (Whittle et al., 2009). In this context, we provide an integrated approach to achieve this objective. We have used two approaches for defining and modeling requirements, i.e., Goal-Oriented Requirements Engineering (GORE) techniques are used to define and model the requirements of SAS (Goldsby et al., 2008; Lapouchnian et al., 2005; Yu et al., 2008; 2004) and SysML is used to specify the system and to provide a link with the requirements.

We propose a model-based requirements modeling and verification process for SAS that takes into account the uncertainty in requirements of these systems. We provide some tools to implement our approach and then apply it on an academic case study. The notion of goals is added to take into account the advantages offered by GORE. Requirements verification is done using a model checking technique.

This paper is organized as follows: In Section 2, we describe the background and the concepts which form the basis of this work, Section 3 shows the state of the art regarding RE for SAS and properties verification of these systems, Section 4 illustrates our proposed approach through an example and the tools that we have developed, Section 5 shows the case study that we used for the validation of our approach, and Section 6 concludes the paper and shows the future work.

2. Background

2.1. RELAX

Relax is an RE language for DASs in which explicit constructs are included to handle uncertainty. For example, the system might wish to temporarily Relax a non-critical requirement in order to ensure that critical requirements can still be met. The need for DASs is typically due to two key sources of uncertainty. First is the uncertainty due to changing environmental conditions, such as sensor failures, noisy networks, malicious threats, and unexpected (human) input; the term *environmental uncertainty* is used to capture this class of uncertainty. A second form of uncertainty is *behavioral uncertainty*, which refers to situations where the requirements themselves need to change. It is difficult to know all requirements changes at design time and, in particular, it may not be possible to enumerate all possible alternatives (Whittle et al., 2009).

2.1.1. RELAX vocabulary

The vocabulary of Relax is designed to enable the analysts to identify the requirements that may be Relax-ed when the environment changes. Relax addresses both types of uncertainties. Relax also outlines a process for translating traditional requirements into Relax requirements. The only focal point is for the requirement engineers to identify the point of flexibility in their requirements. Relax identifies two types of requirements: one that can be Relax-ed in favor of other ones, called variant or Relax-*ed*, and other that should never change, called *invariant*. It is important to note that the decision of whether a requirement is invariant or not is an issue for the system stakeholders, aided by the requirements engineers.

Relax takes the form of a structured natural language, including operators designed specifically to capture uncertainty (Whittle et al., 2008); their semantics is also defined. Fig. 1 shows the set of Relax operators, organized into modal, temporal, ordinal operators and uncertainty factors. The conventional modal verb *SHALL* is retained for expressing a requirement, with Relax operators providing more flexibility in how and when that functionality may be delivered. More specifically, for a requirement that contributes to the satisfaction of goals that may be temporarily left unsatisfied, the inclusion of an alternative, temporal or ordinal Relax-ation modifier, will define the requirement as Relax-able.

RELAX operator	Description
Modal Operators	
SHALL	a requirement must hold
MAY ... OR	a requirement specifies one or more alternatives
Temporal Operators	
EVENTUALLY	a requirement must hold eventually
UNTIL	a requirement must hold until a future position
BEFORE, AFTER	a requirement must hold before or after a particular event
IN	a requirement must hold during a particular time interval
AS EARLY, LATE AS POSSIBLE	a requirement specifies something that should hold as soon as possible or should be delayed as long as possible
AS CLOSE AS POSSIBLE TO [frequency]	a requirement specifies something that happens repeatedly but the frequency may be relaxed
Ordinal Operators	
AS CLOSE AS POSSIBLE TO [quantity]	a requirement specifies a countable quantity but the exact count may be relaxed
AS MANY, FEW AS POSSIBLE	a requirement specifies a countable quantity but the exact count may be relaxed
Uncertainty Factors	
ENV	defines a set of properties that define the system's environment
MON	defines a set of properties that can be monitored by the system
REL	defines the relationship between the **ENV** and **MON** properties
DEP	identifies the dependencies between the (relaxed and invariant) requirements

Fig. 1. Relax operators (Whittle et al., 2009).

$$\begin{aligned}\varphi := & \ true \mid false \mid p \mid SHALL\ \varphi \\ & \mid MAY\ \varphi_1 OR\ MAY\ \varphi_2 \\ & \mid EVENTUALLY\ \varphi \mid \varphi_1 UNTIL\ \varphi_2 \\ & \mid BEFORE\ e\ \varphi \mid AFTER\ e\ \varphi \mid IN\ t\ \varphi \\ & \mid AS\ CLOSE\ AS\ POSSIBLE\ TO\ f\ \varphi \\ & \mid AS\ CLOSE\ AS\ POSSIBLE\ TO\ q\ \varphi \\ & \mid AS\ \{EARLY, LATE, MANY, FEW\} \\ & AS\ POSSIBLE\ \varphi\end{aligned}$$

Fig. 2. Relax grammar (Whittle et al., 2009).

2.1.2. RELAX grammar

The syntax of Relax expressions is defined by the grammar shown in Fig. 2. Parameters of Relax operators are typed as follows: p is an atomic proposition, e is an event, t is a time interval, f is a frequency and q is a quantity. An event is a notable occurrence that takes place at a particular instant in time. A time interval is any length of time bounded by two time instants. A frequency defines the number of occurrences of an event within a given time interval. If the number of occurrences is unspecified, then it is assumed to be one. A quantity is something measurable, meaning it can be enumerated. In particular, a Relax expression φ is said to be quantifiable if, and only if, there exists a function Δ such that $\Delta(\varphi)$ is a quantity. A valid Relax expression is any conjunction of statements $s_1, \ldots, s_m$, where each s_i is generated by the grammar.

The semantics of Relax expressions is defined in terms of Fuzzy Branching Temporal Logic (FBTL) (Moon et al., 2004). FBTL can describe a branching temporal model with uncertain temporal and logical information. It is the representation of uncertainty in FBTL that makes it suitable as a formalism for Relax.

2.1.3. RELAX process

Fig. 3 shows the Relax process. The conventional process of requirement discovery has been applied to get *SHALL* statements. Relax process is then used to identify the requirements as invariant and Relax-ed.

First of all, for each *SHALL* statement, we check whether it must always be satisfied or not. Then for each potentially Relax-able requirement, we identify the uncertainty factors. Here also the observable properties of the environment are identified. The *ENV*/*MON* relationship is made explicit by *REL*, and *DEP* is used to identify the inter-dependencies between requirements. Then we check whether the *SHALL* statement should be Relax-ed to handle uncertainty factors or not. Here we analyze the uncertainty factors to determine if sufficient uncertainty exists in the environment that makes absolute satisfaction of the requirement problematic or undesirable. If so, then this *SHALL* statement needs to proceed to the next step for introducing Relax operators. If, however, the analysis reveals no uncertainty in its scope of the environment, then the requirement is potentially always satisfiable and therefore identified as an invariant.

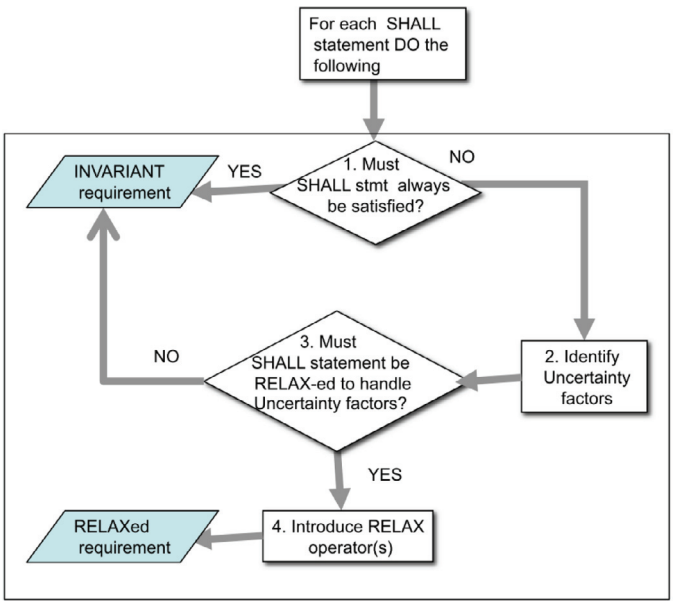


Fig. 3. Relax process (Whittle et al., 2009).

After the application of Relax process on traditional requirements, we obtain invariant and Relax-ed requirements. Relax-ed requirements support a high degree of flexibility that goes well beyond the original requirements. Once the requirements engineer determines that indeed a level of flexibility can be tolerated, then the downstream developers, including the designers and programmers, have the flexibility to incorporate the most suitable adaptive mechanisms to support the desired functionality. These decisions may be made at design time and/or runtime (Blair et al., 2009; Cheng et al. , 2009b).

2.2. SysML/KAOS

The SysML/Kaos (Gnaho and Semmak, 2010) model is an extension of the SysML[1] requirements model, with concepts of the Kaos goal model (Lamsweerde, 2009). SysML is an extension of Uml,[2] so it provides concepts to represent requirements and to relate them to other model elements, allowing the definition of traceability links between requirements and system models. The SysML/Kaos metamodel is implemented as a new profile, importing the SysML profile.

2.2.1. SysML

SysML is a general purpose modeling language for systems engineering applications. SysML is a Uml profile that represents a subset of Uml 2.0 with extensions. It supports the specification, analysis, design, verification and validation of a broad range of systems and systems-of-systems. These systems may include hardware, software, information, processes, personnel, and facilities. In particular, the language provides graphical representations with a semantic foundation for modeling system requirements, behavior, structure, and constraints, which is used to integrate with other engineering analysis models.

SysML includes a graphical construct to represent text-based requirements and relate them to other model elements. The requirements diagram captures requirements hierarchies and requirements derivation, and the <<satisfy>> and <<verify>> relationships allow a modeler to relate a requirement to a model element, e.g., <<block>>, that satisfies or verifies the requirements. The requirement diagram provides a bridge between typical requirements management tools and system models.

2.2.2. KAOS

Kaos is a goal-oriented methodology for RE, enabling analysts to build requirements models and to derive requirements documents from Kaos models. The first key idea behind Kaos is to build a model for the requirements, i.e., for describing the problem to be solved and the constraints that must be fulfilled by any solution provider. Kaos has been designed: (i) To fit problem descriptions by allowing to define and manipulate concepts relevant to problem description ; (ii) To improve the problem analysis process by providing a systematic approach for discovering and structuring requirements ; (iii) To clarify the responsibilities of all the project stakeholders ; (iv) To let the stakeholders communicate easily and efficiently about the requirements.

[1] http://www.omgsysml.org/
[2] http://www.omg.org/spec/UML/

2.2.3. Why SysML/KAOS?

SysML and Kaos have some advantages and weak points, but these are complementary to each other based on the following points: (i) Requirements description: A textual description in SysML and a description in the form of goals in Kaos ; (ii) Relation between requirements: SysML has <<*contain*>> and <<*derive*>> relations; these relations do not have precise semantics, which leads to confusion. Kaos has refinement relations AND/OR ; (iii) Traceability relations: <<*satisfy*>> and <<*verify*>> relations in SysML allow to define traceability. Kaos does not have explicit traceability relations ; (iv) Tools: A number of tools exist for SysML; most of them are open source. Kaos propose a proprietary tool called Objectiver.[3]

Traditionally, requirements are divided into Functional Requirements (FRs) and Non-Functional Requirements (NFRs). Due to the complexity of systems, NFRs should be processed much earlier than when they are usually handled in most development processes, at the same level of abstraction as FRs which will allow taking into account these properties for the evaluation of alternate options, risk and conflict analysis. The benefit of SysML is that it allows throughout the development cycle to relate requirements to other model elements, thus ensuring continuity from the requirements phase to the implementation phase. However, the proposed concepts of requirements in SysML are not as rich as in the other RE methods (especially GORE). SysML/Kaos is the result of motivation to benefit from the contributions of SysML, while ensuring a more precise definition of the concepts. SysML/Kaos is inspired from the work of Chung et al. (1999) and Cysneiros and Leite (2004). The SysML/Kaos model allows both FRs (Laleau et al., 2010) and NFRs (Gnaho and Semmak, 2010) to be modeled.

2.2.4. SysML/KAOS meta-model

Fig. 4 shows the extended meta-model of SysML/Kaos (Gnaho and Semmak, 2010); non-functional concepts are represented as yellow boxes (bottom), the gray boxes (top) represent the SysML concepts. The instantiation of the meta-model allows us to obtain a hierarchy of NFRs in the form of goals. Non-Functional Goals (NFGs) are organized in refinement hierarchies. The meta-class NonFunctionalGoal represents the Non-Functional Goal (NFG), it is specified as a subclass of the meta-class Goal, which itself is a subclass of the meta-class Requirement of SysML. An NFG represents a quality that the future system must have. The nFGType specifies the type of NFG and the attribute topic represents the domain concept concerned by this type of requirement. An NFG can thus be represented with the following syntax: nFGType [topic]. An NFG is either an AbstractNFG or an ElementaryNFG. A goal that cannot be further refined is an ElementaryNFG. The refinement of an AbstractNFG by either abstract or elementary goals is represented by the AssociationClass Refinement. An AbstractNFG may contain several combinations of subgoals (abstract or elementary). The relationship Refinement becomes an AssociationClass between an AbstractNFG and its subgoals. It can be specialized to represent And/Or goal refinements. At the end of the refinement process, it is necessary to identify and express the various alternative ways to satisfy the ElementaryNFGs. For that, the SysML/Kaos meta-model considers the concept of the meta-class ContributionGoal. A ContributionGoal captures a possible way to satisfy an ElementaryNFG. The AssociationClass Contribution describes the characteristics of the contribution. It provides two properties: contributionNature and contributionType. The first one specifies whether the contribution is *positive* or *negative*, whereas the second one specifies whether the contribution is *direct* or *indirect*. A *positive* (resp. *negative*) contribution helps positively (resp. negatively) to the satisfaction of an ElementaryNFG. A *direct contribution* describes an explicit contribution to the ElementaryNFG. An *indirect contribution* describes a kind of contribution that is a direct contribution to a given goal but induces an unexpected contribution to another goal. Finally, the concept of *Impact* is used to connect NFGs to Functional Goals (FGs). It captures the fact that a ContributionGoal has an effect on FGs.

[3] http://www.objectiver.com/

2.3. The OMEGA2 UML/SysML profile and IFx toolset

Formal methods provide tools to verify the consistency and correctness of a specification, with respect to the desired properties of the system. For this reason, we use these methods to prove some of the properties of the system before the system development even starts. We use OMEGA2/IFx profile and toolset for the properties verification and model simulation of our case study.

2.3.1. The OMEGA2 Profile

OMEGA2 profile (Ober and Dragomir, 2010) is an executable Uml/SysML profile used for the formal specification and validation of critical real-time systems. It is based on a subset of Uml 2.2/SysML 1.1 containing the main constructs for defining the system structure and behavior.

The OMEGA2 Uml/SysML profile defines the semantics of Uml/SysML elements providing the means to model coherent and unambiguous system models. In order to make the models verifiable, it presents as extension the *observers* mechanism for specifying dynamic properties of models. The OMEGA2 Uml/SysML Profile is implemented by the IFx toolbox which provides static analysis, simulation and timed automaton-based model-checking (Clarke et al., 1999) techniques for validation.

The architecture of an OMEGA2 model is described in Class/Block Definition Diagrams by classes/blocks with their relationships. Each class/block defines properties and operations, as well as a state machine. The hierarchical structure of a model is defined in composite structures/Internal Block Diagram (IBD): parts that communicate through ports and connectors. For the SysML Block Definition Diagram (BDD), the following concepts are taken into account: blocks and their relationships (association, aggregation, generalization), interfaces, basic types, signals.

For the system behavior, the OMEGA2 profile takes into account the following concepts: State machines (excluding: history states, entry point, exit point, junction) and Actions; for this, the profile defines a concrete syntax. This syntax is used for example to define operation bodies and transition effects in state machines. The textual action language is compatible with the Uml 2.2 action meta-model and implements its main elements: object creation and destruction, operation calls, expression evaluation, variable assignment, signal output, return action as well as control flow structuring statements.

For specifying and verifying dynamic properties of models, OMEGA2 uses the notion of *observers. Observers* are special classes/blocks monitoring run-time state and events. They are defined by classes/blocks stereotyped with <<*observer*>>. They may have local memory (attributes) and a state machine describes their behavior. States are classified as <<*success*>> and <<*error*>> states to express the satisfaction (or not) of safety properties. The main issue in modeling *observers* is the choice of events which trigger their transitions.

The trigger of an *observer* transition is a match clause specifying the type of event (e.g., receive), some related information (e.g., the operation name) and observer variables that may receive related information (e.g., variables receiving the values of operation call parameters). Besides events, an observer may access any part of the state of the Uml model: object attributes and state, signal queues.

2.3.2. IFx toolset

OMEGA2 models can be simulated and properties can be verified using the IFx toolset (Bozga et al., 2004). The IFx toolset

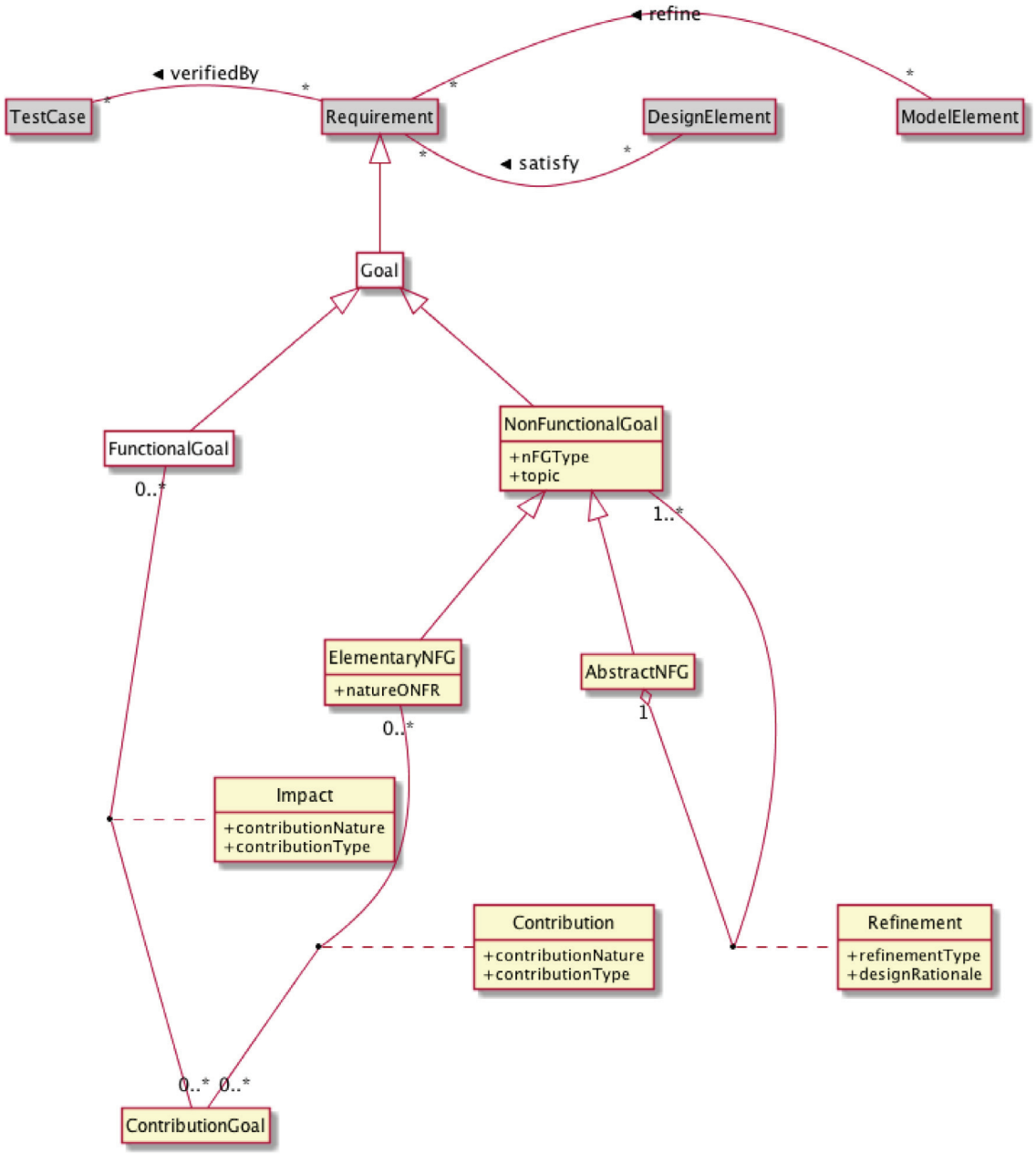


Fig. 4. SysML/Kaos meta model (Gnaho and Semmak, 2010). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

provides verification which ensures the automatic process of verifying whether an OMEGA2 UML/SysML model satisfies (some of) the properties (i.e., *observers*) defined on it. The verification method employed in IFx is based on systematic exploration of the system state space (i.e., enumerative model checking). The IFx toolset also provides simulation which designates the interactive execution of an OMEGA2 UML/SysML model. The execution can be performed step-by-step, random, or guided by a simulation scenario (for example an error scenario generated during a verification activity).

The IFx toolset relies on a translation of UML/SysML models toward a simple specification language based on an asynchronous composition of extended timed automata: the IF language,[4] and on the use of simulation and verification tools available for IF. The translation takes an input model in XML Metadata Interchange (XMI) 2.0 format. The compiler verifies the set of well-formedness rules imposed by the profile and generates an IF model that can be further reduced by static analysis techniques. This model is subject to verification that either validates the model with respect to its properties or produces a list of error scenarios that can be further debugged using the simulator. The OMEGA2/IFx approach has been applied for the verification and validation of industry grade models (Dragomir et al., 2012) providing interesting results.

3. State of the art

Different roadmap papers on Software Engineering (SE) for SAS (Cheng et al., 2009b; Rogério de Lemos et al., 2013) discuss the state of the art, its limitations, and identify critical challenges. Cheng et al. (2009b) present a research roadmap for SE of SAS focusing on four views, which are identified as essential: requirements, modeling, engineering, and assurances. The focus is on development

[4] http://www-if.imag.fr/

methods, techniques, and tools that seem to be required to support the systematic development of complex software systems with dynamic self-adaptive behavior. The most recent roadmap paper (Rogério de Lemos et al., 2013) discusses four essential topics of self-adaptation: design space for self-adaptive solutions, software engineering processes for self-adaptive systems, from centralized to decentralized control, and practical run-time verification and validation for SAS.

3.1. Requirements Engineering for Self-Adaptive Systems

An SAS is able to modify its behavior according to changes in its environment. As such, an SAS must continuously monitor changes in its context and react accordingly. But here the question arises as to what aspects of the environment the SAS should monitor. Clearly, the system cannot monitor everything and exactly what should the system do if it detects a less than optimal pattern in the environment? Presumably, the system still needs to maintain a set of high level goals that should be maintained regardless of the environmental conditions. But non-critical goals could well be RELAX-ed, thus allowing the system a degree of flexibility during or after adaptation. It is important to identify these properties as early as possible.

Levels of Requirement Engineering for Modeling (LoREM) (Goldsby et al., 2008) is an approach for modeling the requirements of Dynamic-Adaptive Systems (DAS) using i* goal models (Yu, 1997). The i* goal models are used to represent the stakeholder objectives, non-adaptive system behavior (business logic), adaptive behavior, and adaptation mechanism needs of DAS. Each of these i* goal models addresses the three RE concerns (conditions to monitor, decision-making procedure, and possible adaptations) from a specific developers perspective.

Awareness Requirements (AwReqs) (Vítor et al., 2011) are requirements that talk about the success or failure of other requirements. More generally, AwReqs talk about the states requirements can assume during their execution at run-time. AwReqs are represented in a formal language and can be directly monitored by a requirements monitoring framework.

CLAIMS (Welsh and Sawyer, 2010; Welsh et al., 2011) were applied as markers of uncertainty to record the rationale for a decision made with incomplete information in DASs. The work in Ramirez et al. (2012a) integrates RELAX and CLAIMS to assess the validity of CLAIMS at run-time while tolerating minor and unanticipated environmental conditions that can otherwise trigger adaptations.

RELAX can be used in goal oriented modeling approaches for specifying and mitigating sources of uncertainty in DASs (Cheng et al., 2009a). AutoRELAX (Ramirez et al., 2012b), is an approach that generates RELAX-ed goal models that address environmental uncertainty by identifying which goals to RELAX, which RELAX operators to apply, and the shape of the fuzzy logic function that defines the goal satisfaction criteria. AutoRELAX also requires an executable specification of the DAS, such as a simulation or a prototype, which applies the set of utility functions to measure how well the DAS satisfies its requirements in response to adverse conditions. For the experimental setup of AutoRELAX, a null hypothesis is defined which states that there is no difference between a RELAX-ed and an unRELAX-ed goal model.

Fuzzy Live Adaptive Goals for Self-Adaptive Systems (FLAGS) (Baresi et al., 2010) is an innovative goal model which deals with the challenges posed by SAS. Goal models have been used for representing systems requirements, and also for tracing them onto their underlying operationalization.

The state of the art regarding RE for SAS shows different approaches from the point of view of its complementarity with RELAX. The different steps in LoREM are interesting but our focus is on RELAX-ed requirements as we want to identify the uncertainty in the requirements of DASs. Regarding AwReqs, in future work, we want to integrate this concept into our approach using Monitor-Analyze-Plan-Execute (MAPE) (Kephart and Chess, 2003) feedback loop that operationalizes the system's adaptability mechanisms. CLAIMS are also subject to uncertainty, in the form of unanticipated environmental conditions and unreliable monitoring information, that can adversely affect the behavior of the DAS if it spuriously falsifies a claim. A CLAIM can also be monitored at runtime to prove or disprove its validity (Welsh et al., 2011), thereby triggering adaptation to reach more desirable system configurations if necessary. CLAIMS therefore complement RELAX.

3.2. Properties verification of SAS

For the properties verification of SAS, we use the OMEGA2/IFx profile and toolset which was developed in our team (Ober and Dragomir, 2010). The advantage of the OMEGA2 profile is that it provides the notion of *observers* for specifying and verifying dynamic properties of models. In terms of properties verification, there exists a number of techniques. In the following, we give a description of some of it.

Benghazi et al. (2009) present a verification approach based on MEDISTAM-RT, which is a methodological framework for the design and analysis of real-time systems and timed traces semantics, to check the fulfillment of NFRs. It only focuses on safety and timeliness properties, to assure the correct functioning of Ambient Assisted Living (AAL) systems and to show the applicability of this methodology in the context of this kind of system.

Apvrille et al. (2004) introduce a profile named Timed UML and RTLOTOS Environment (TURTLE) which extends the UML class and activity diagrams with composition and temporal operators. TURTLE is a real-time UML profile with a formal semantics expressed in Real-Time Language Of Temporal Ordering Specifications (RTLOTOS) (Courtiat et al., 2000). With its formal semantics and toolkit, TURTLE enables a priori detection of design errors through a combination of simulation and verification/validation techniques.

In Laleau et al. (2010), the authors propose an extension to SysML with concepts from the goal model of the KAOS method (SysML/KAOS) with rules to derive a formal B (Abrial, 1996) specification from this goal model. The B formal method is a complete method that supports a large segment of the software development life cycle: specification, refinement and implementation.

In MEDISTAM-RT, the focus is on safety and timeliness properties, we do not treat any specific type of properties. We verify those requirements that are of interest for adaptation in SAS. In TURTLE, design errors can be detected through simulation and verification. That is the reason why we plan to explore the complementarity of this approach with our approach. The use of formal methods like B can help avoid the state space explosion problem which is inherent in model checking techniques. We have worked on studying the complementarity of these two approaches and we plan to integrate them in our approach in the future work.

4. Proposed approach

In this section, we introduce the overall view of our proposed approach (Ahmad, 2013). We show our contribution then we describe the overall process of our approach. To illustrate our proposed approach, we use requirements from the barbados Car Crash Crisis Management System (bCMS) case study. At the end, we show the integrated tooling environment that we developed to validate our approach.

4.1. Contribution

To properly define the scope of our contribution, it is necessary to identify the work we have done. Firstly, we have found that although the use of traditional process of SysML/KAOS was interesting for modeling the requirements of SAS, it does not take into account

the notion of uncertainty. On the other hand, RELAX is a process tailored to identify and highlight the uncertainty, but it does not provide tools for its implementation. Finally, the verification techniques used for these models do not take into account the uncertainty posed by these systems. Based on this observation, we contributed toward the definition of an integrated tool-based process. For this, we developed support for RELAX. Then we developed rules to transform requirements addressed by RELAX to SYSML/KAOS, using model transformation techniques. Finally, we integrated formal verification techniques i.e., OMEGA2/IFx in the process. To reduce the risk of state space explosion problem (Clarke et al., 2012) when we take into account the whole system using OMEGA2/IFx, we limited its use to verify only adaptable properties. We present in detail the work and the overall process in the next section.

4.2. The proposed approach

In the following, each step of the proposed approach is explained with associated input and output. Fig. 5 shows the overall view of our proposed approach.

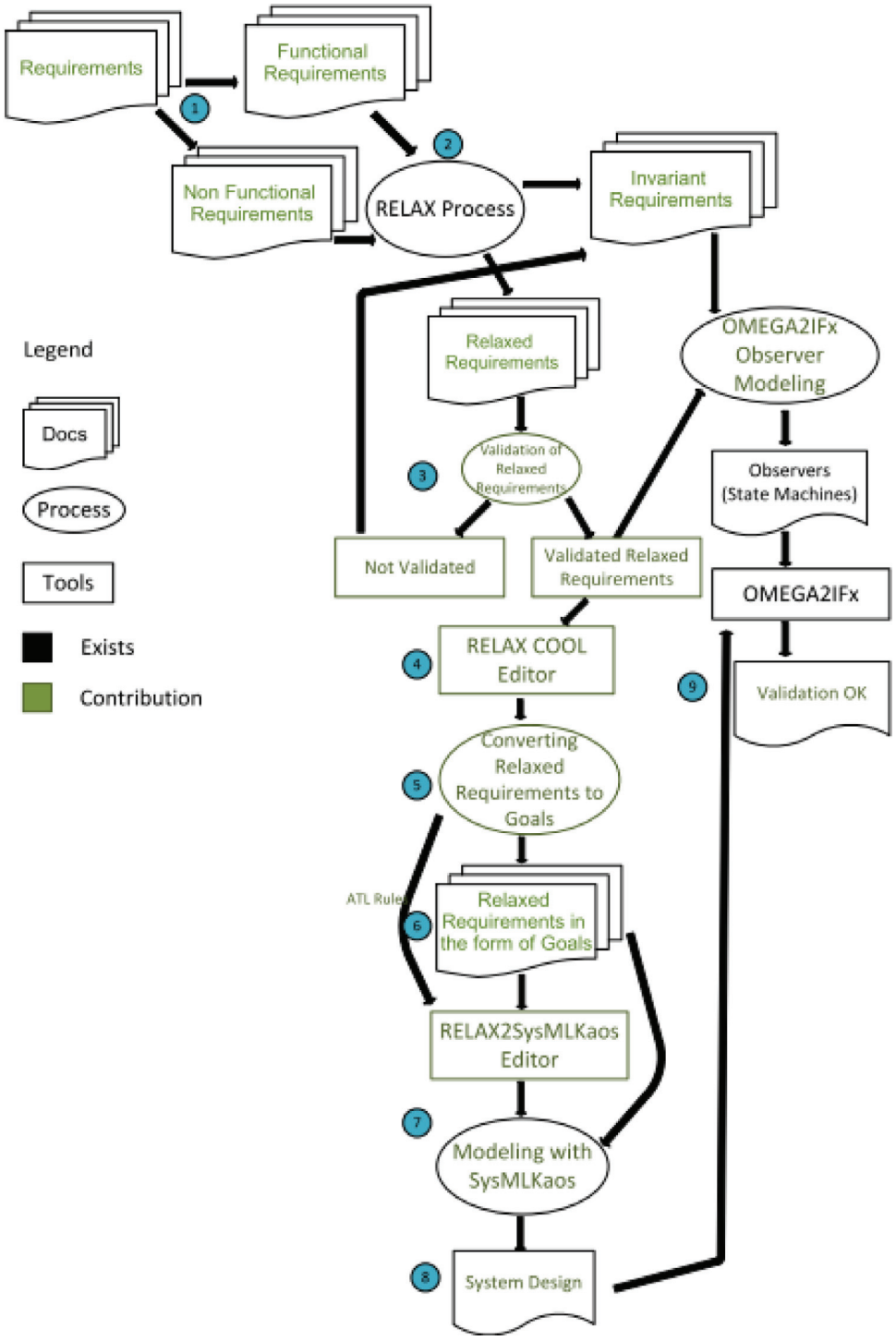


Fig. 5. Overall view of our approach.

1. The overall approach that we propose takes requirements as input. These requirements are elicited in the form of *SHALL* statements by a requirement engineer which are then divided into FRs and NFRs.
2. We apply RELAX process (see Section 2.1.3) on these FRs and NFRs to get those requirements that are associated with the adaptability features of SAS called RELAX-ed requirements and those that are fixed called invariant requirements.
3. Here, we validate the RELAX-ed requirement with the help of an expert i.e., for each RELAX-ed property, we check whether the new expression of the property is acceptable or not. By *acceptable* we mean two things: (i) the RELAX-ed expression is sound (it can be operationalized), and (ii) the boundaries make sense (from the domain expert point of view). If the RELAX-ed expression is acceptable then we proceed with the next step, if it is not acceptable, we propose two options: cancel the RELAX-ation and go back to a *SHALL* invariant or complement the RELAX-ed property with an additional invariant (e.g., a *max* or *min* boundary that constraints the RELAX-ed expression).
4. The resulting RELAX-ed requirements are then formalized using an editor that we developed called RELAX COOL editor. This editor takes into account the uncertainty factors associated with each RELAX-ed requirement. Xtext[5] is used for the development of this editor.
5. At this point, we use a process for the conversion of RELAX-ed requirements into goal concepts i.e., SysML/KAOS. We use a correlation table (see Section 4.3.1) for the correspondence between RELAX-ed requirements and SysML/KAOS concepts (Ahmad et al., 2012b). For this purpose, we have developed a tool called RELAX2SysML/KAOS editor, which is based on Atlas Transformation Language (ATL) transformations. For the time being, the tool helps in mapping the RELAX concepts to SysML/KAOS concepts but not the inverse.
6. At this step, we have a full list of RELAX-ed requirements with uncertainty factors converted into SysML/KAOS goal concepts.
7. The non-functional RELAX-ed requirements in the form of SysML/KAOS goal concepts can now be modeled with the help of SysML/KAOS editor.
8. This step shows the system design. The RELAX-ed requirements of the SAS are now modeled and we have a snapshot of the system design.
9. Once we have the system design, we use the OMEGA2/IFx *observers* to verify the properties of SAS. The input to this step are the OMEGA2/IFx *observers* which are the RELAX-ed and invariant requirements. The verification either results in the fulfillment of all the properties or if there is an error produced during verification, it can be simulated through the interactive simulation interface of the IFx toolset in order to identify the source of the error and then subsequently correct it in the model.

4.3. Integration of the approaches

In the following, we present how we defined the convergence between different methods used in our approach.

4.3.1. Relationship between RELAX, SysML/KAOS and SysML

In our integrated approach, we take benefit of SysML/KAOS while modeling RELAX-ed requirements of SAS. In Fig. 6, we show how several key concepts are taken into account in the selected approaches. The concepts are taken from RELAX and are then compared with the other approaches.

- In SysML/KAOS, requirements are described in the form of goals; SysML describes requirements in textual form; RELAX requirements are also in textual form which contains more information in the form of RELAX operators.
- To deal with monitoring, SysML/KAOS has the *Contribution Goal* concept which is used to satisfy an *Elementary NFG*, SysML has *<<satisfy>>* which is used when a *<<block>>* satisfies a *<<requirement>>* while for RELAX, we have the concept of *MON* which is used to measure the environment, i.e., *ENV*.
- SysML/KAOS has the concept of *Contribution* which is an *Association Class* between *Contribution Goal* and *Elementary NFG*. *Contribution* describes the characteristics of the contribution. It provides two properties: *ContributionNature* and *ContributionType*. SysML has *<<verify>>* and *<<refine>>* relationships while for RELAX, we have *REL* variable which identifies the relationship between *ENV* and *MON* or more precisely how *MON* achieves *ENV*.
- For Dependency/Impact, SysML/KAOS describes it as an *Impact* of a *Contribution Goal* on a Functional Goal (FG). It also has the same two properties, i.e., *ContributionNature* and *ContributionType*. This impact can be *positive* or *negative* and *direct* or *indirect*. In SysML, we have the concept of *<<derive>>* which shows the dependency between requirements, RELAX has *positive* and *negative* dependency which shows the dependency of a RELAX-ed requirement on other requirements.
- For the tools available for each approach, SysML/KAOS has a tool called SysML/KAOS editor, SysML has a number of tools e.g., eclipse,[6] Papyrus,[7] topcased,[8] etc. and for RELAX, we have developed an eclipse-based RELAX COOL editor (Bascans et al., 2013). We have also developed RELAX2SysML/KAOS editor which does the mapping between RELAX uncertainty factors and SysML/KAOS goal concepts.

4.3.2. Uncertainty factors/impacts

SysML/KAOS is a GORE approach that takes into account different kinds of dependencies between *Goals* and *Contribution Goals*. RELAX deals with dependency in terms of the dependency of a RELAX-ed requirement on an invariant requirement but it does not say anything about the dependency of a *Monitor* (*Contribution Goal* in SysML/KAOS) on *ENV* (*Goal* in SysML/KAOS). So the injection of SysML/KAOS in our approach helps in capturing the dependencies between different requirements and also between the monitors and environment. RELAX uses a kind of vocabulary that only captures uncertainty in the requirements of SAS while KAOS helps in allowing the stakeholders communicate easily and efficiently about requirements.

RELAX uncertainty factors, especially *ENV* and *MON*, are particularly important for documenting whether the system has means for monitoring the important aspects of the environment. By collecting these *ENV* and *MON* attributes, we can build up a model of the environment in which the system will operate, as well as a model of how the system monitors its environment. In RELAX, requirements dependencies are delimited by the uncertainty factor *DEP*, as it is important to assess the impact on dependent requirements after RELAX-ing a given requirement. Having said this, SysML/KAOS can complement RELAX by injecting more information in the form of *positive/negative* and *direct/indirect* impacts (Ahmad et al., 2012a), which models the impact of a *Contribution Goal* on an *Elementary Goal*. The grammar of RELAX acts as a meta-model for our RELAX COOL editor, while SysML/KAOS has extended the meta-model of SysML with goal concept. As both meta-models are close to the SysML meta-model, we have bridged RELAX and SysML/KAOS using our proposed approach.

[5] http://www.eclipse.org/Xtext/

[6] http://www.eclipse.org/

[7] http://www.papyrusuml.org

[8] http://www.topcased.org/

Concepts/Approaches	SysML/KAOS	SysML	RELAX
Requirements Description	AbstractGoal ElementaryGoal	Textual Requirements	Relaxed Requirement ENV
Monitoring	Contribution Goal	<<satisfy>>	MON
Relationship	Contribution Nature: Positive Negative Contribution Type: Direct (Explicit) Indirect (Implicit)	<<verify>> <<refine>>	REL
Dependency/Impact	Contribution Nature: Positive Negative Contribution Type: Direct (Explicit) Indirect (Implicit)	<<derive>> <<contain>>	DEP: Positive Negative
Tools	Eclipse based SysML/KAOS Editor	Eclipse/Papyrus/Topcased/ …..	Eclipse based COOL RELAX editor

Fig. 6. Relationship b/w SysML/Kaos SysML and Relax.

4.3.3. Verification of ambient system's properties through formal methods

Using our proposed approach, we provide a strong consistency between models. This can be ensured thanks to the use of formal methods that provide verification tools for the properties verification and model simulation of SAS. We have integrated OMEGA2/IFx for properties verification and model simulation of these systems in our proposed approach. By doing this, we bridge the gap between the requirements phase and the initial formal specification phase.

4.4. Proposed approach illustration

To illustrate our approach, we use the bCMS [9] case study. Here is an excerpt of the case study.

The bCMS is a distributed crash management system that is responsible for coordinating the communication between a Fire Station Coordinator (FSC) and a Police Station Coordinator (PSC) to handle a crisis in a timely manner. Information regarding the crisis as it pertains to the tasks of the coordinators is updated and maintained during and after the crisis. There are two collaborative sub-systems. Thus, the global coordination is the result of the parallel composition of the (software) coordination processes controlled by the two (human) distributed coordinators. There is no central database; fire and police stations maintain separate databases and may only access information from the other database through the bCMS system. Each coordination process is hence in charge of adding and updating information in its respective database. Fig. 7 shows the overall view of the bCMS case study.

We have chosen an (illustrative) subset of the bCMS requirements. The requirements are numbered in a shared document.[10]

We have first applied the Relax process on bCMS requirements to get invariant and Relax-ed requirements. For Relax-ed requirements, all the uncertainty factors were identified. Then using the correlation in Fig. 6, we have modeled the bCMS system requirements with the SysML/Kaos approach. In Ahmad et al. (2013a), we have modeled some more requirements of the bCMS case study. Following are some of the Relax-ed requirements that we identified:

- Relax-ed requirements: R4, R8.

Fig. 8 shows the uncertainty factors associated with the *Integrity* R4 (*The system shall ensure that the integrity of the communication between coordinators regarding crisis location, vehicle number, and vehicle location is preserved AS CLOSE AS POSSIBLE TO 99.99% of the time.*) Relax-ed requirement.

Fig. 9 shows the uncertainty factors associated with the *Availability* R8 (*The crisis details and route plan of the fire station and the police station shall be available with the exception of AS CLOSE AS POSSIBLE To* 0 *minutes AND* $\leq$ 30 *min for every 48 h when no crisis is active.*) Relax-ed requirement.

Fig. 10 shows a low level goal model of the bCMS case study. We have identified a goal *"Ensure the integrity of communications b/w*coordinators[bCMS]"*which is an abstract non-functional goal and*

[9] Available at http://cserg0.site.uottawa.ca/cma2013re/CaseStudy.pdf.

[10] Available at http://goo.gl/uscP5

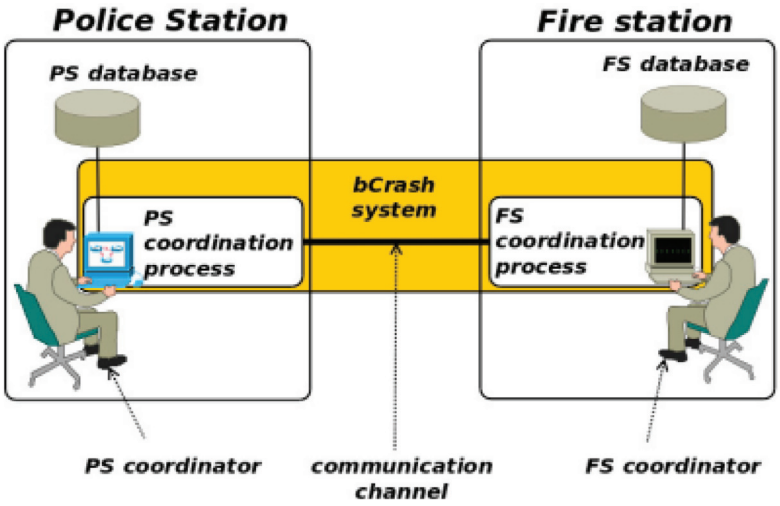


Fig. 7. bCMS case study overall view.

Uncertainty Factors	Details
ENV	Integrity of the communication between coordinators, Authenticity of the coordinators to avoid the communication compromiser
MON	Secure communication channel, use PIN code, use Additional information, Communication Compromiser
REL	Secure communication channel ensures the integrity of the communication between coordinators, PIN code and Additional information ensures that the authenticity of the coordinators is in place, The communication compromiser compromises the integrity of coordinators

Fig. 8. R4 integrity RELAX-ed requirement uncertainty factors.

Uncertainty Factors	Details
ENV	The crisis details and route plan of the fire station shall be available, the crisis details and route plan of the police station shall be available
MON	Fire Station Coordinator, Police Station Coordinator, Communication Compromiser
REL	Fire Station Coordinator updates the crisis details and route plan of the fire station, Police Station Coordinator updates the crisis details and route plan of the police station, The Communication Compromiser compromises the availability of data

Fig. 9. R8 availability RELAX-ed requirement uncertainty factors.

is AND-refined into two sub-goals using refinement by type: (i) Integrity of communication b/w coordinators[bCMS] and (ii) *Authenticity of coordinators[bCMS]*. The goal *Integrity of communication b/w coordinators[bCMS]* is satisfied by the *Contribution Goal Secure communication channel*. Considering the goal *Authenticity of coordinators[bCMS]*, one possible way to achieve this goal is to use *PIN code*, another solution is to use *additional information*. The *Contribution Goal Communication Compromiser* has a *direct* and *negative* impact on the goal *Integrity of communication b/w coordinators[bCMS]*. The functional goal R3: *A PSC maintains control over a crisis situation by communicating with the FSC as well as policemen.* This goal is AND-refined into two sub-goals: *To provide coordinated route plan* and *To estimate resources*. The *Contribution Goal Communication Compromiser* has an *indirect* and *negative* impact on the functional goal *To estimate resources*. The property verification part of our proposed approach is illustrated in Section 5.2.1.

4.5. Tools support

In this section, we introduce the tools that implements our proposed approach.

4.5.1. RELAX editor

For the generation of RELAX editor, Xtext is used. Xtext is a framework for the development of Domain Specific Languages (DSL) and other textual programming languages and helps in the development of an Integrated Development Environment (IDE) for the DSL. Some of the IDE features that are either derived from the grammar or easily implementable are: syntax coloring, model navigation, code completion, outline view, and code templates. An initial version of the RELAX editor can be found in Ahmad (2010). The RELAX grammar is used as a meta-model for this editor which is generated by Xtext that we call RELAX.ecore. Fig. 11 shows an example of the RELAX file with

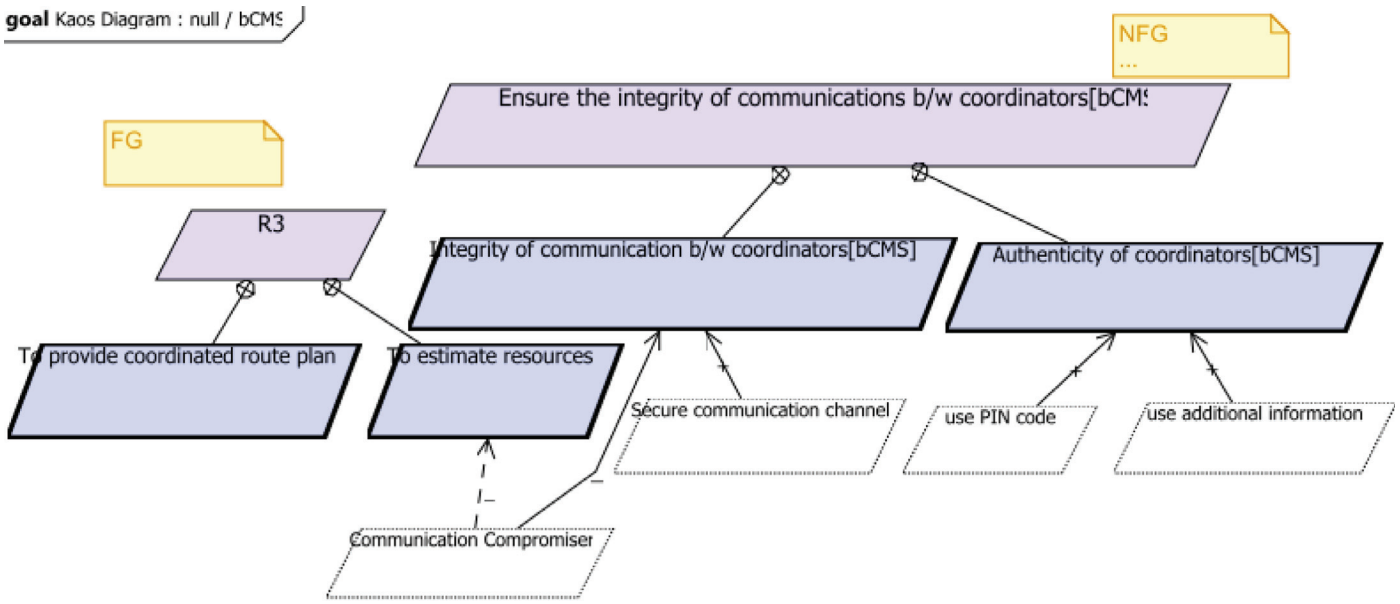


Fig. 10. Low level goal model.

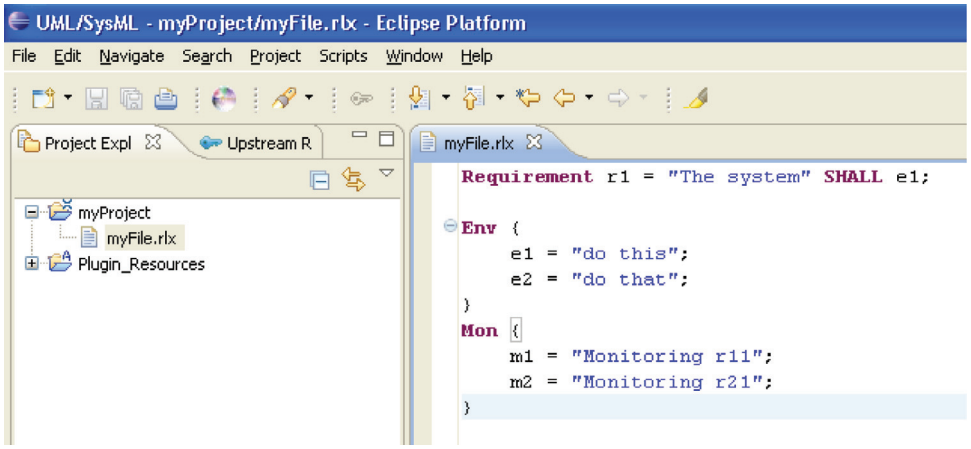


Fig. 11. RELAX file.

uncertainty factors. The RELAX file is represented with an extension *.rlx*. Once we have the *.rlx* file, we can transform it into an XMI model. The XMI model can then be manipulated and will serve us for the model transformation from RELAX to SYSML/KAOS as explained in the next section.

4.5.2. RELAX to SysML/Kaos transformation

In our approach, we want to transform RELAX-ed requirements uncertainty factors into SYSML/KAOS goal concepts. This transformation will help in taking into account the adaptability features associated with SAS in the form of uncertainty factors of RELAX-ed requirements and then modeling these requirements in SYSML/KAOS. In this way, we can benefit from the advantages offered by GORE. For this purpose, the RELAX and SYSML/KAOS meta-models are used.

4.5.3. ATL rules

ATL is a model transformation language and toolkit. It provides a way to produce a number of target models from a set of source models. An ATL transformation program is composed of rules that define how source model elements are matched and navigated to create and initialize the elements of the target models. The generation of target model elements is achieved through the specification of transformation rules.

4.5.4. Mapping between RELAX and SysML/Kaos elements

Here, we present the relationship between RELAX and SYSML/KAOS elements. The RELAX abstract syntax is defined in the RELAX meta-model. In turn, the SYSML/KAOS abstract syntax is defined in the SYSML/KAOS meta-model.

Fig. 6 shows the mapping between the two concepts. For the ATL transformation rules, a RELAX-ed requirement is mapped to an *Abstract Goal* as shown in Fig. 12, an *ENV* is mapped to an *Elementary Goal* and *MON* is mapped to *Contribution Goal*. Fig. 13 shows the generated SYSML/KAOS model after the application of ATL rules. Fig. 14 shows the SYSML/KAOS model opened in the editor.

5. Proof of concepts

In this section, we apply our approach on an academic AAL case study. The goal of AAL solutions is to apply ambient intelligence technology to enable people with specific demands, e.g., handicapped or elderly, to live in their preferred environment (Benghazi et al., 2009). In order to achieve this goal, different kinds of AAL systems can be proposed and most of them pose reliability issues and describe important constraints upon the development of software systems (Cleland-Huang et al., 2007). We model the requirements of an

```
rule RelaxedRequirement2AbstractGoal {
        from
                relaxedRequirement : RelaxMetaModel!RelaxedRequirementDeclaration
        to
                abstractGoal : SysMLKAOSMetaModel!AbstractGoal (name <- relaxedRequirement.name)
}
```

Fig. 12. Relaxed requirement to abstract goal mapping.

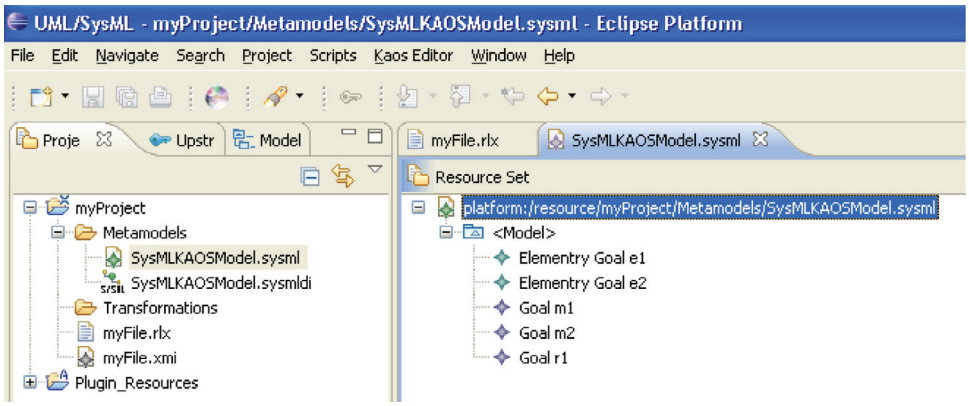


Fig. 13. SysML/Kaos model.

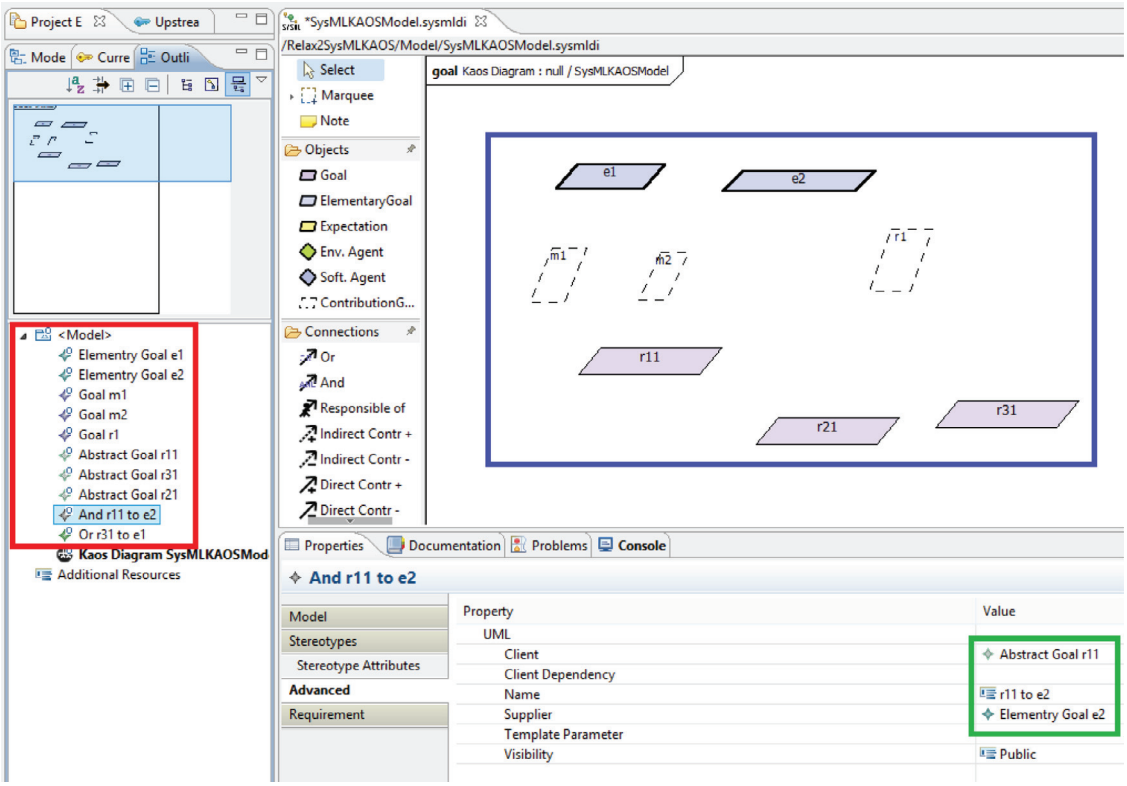


Fig. 14. Generated SysML/Kaos model using ATL transformations.

AAL[11] home which ensures the health of a *Patient* like the one studied by research teams at the IUT of Blagnac.[12] We then show the verification of some of the properties of the AAL system.

5.1. Requirements modeling of the AAL case study

Fig. 15 shows an excerpt of the case study which highlights the need to ensure *Patient*'s health in the AAL home. Advanced smart homes, such as Mary's AAL, rely on adaptivity to work properly. For example, the sensor-enabled cups may fail, but since maintaining a minimum of liquid intake is a life-critical feature, the AAL should be able to respond by achieving this requirement in some other way (Whittle et al., 2009).

Fig. 16 shows an example of RELAX-ed requirement from the Mary's AAL home, which results from the application of the RELAX process on the traditional requirement: *The Fridge shall read, store and communicate RFID information on food packages*. Ahmad (2014) shows the application of RELAX process on some of the requirements of the AAL case study.

[11] http://www.iese.fraunhofer.de/fhg/iese/projects/med_projects/aal-lab/index.jsp
[12] http://mi.iut-blagnac.fr/

Mary is a widow. She is 65 years old, overweight and has high blood pressure and cholesterol levels. Mary gets a new intelligent fridge. It comes with 4 temperature and 2 humidity sensors and is able to read, store, and communicate RFID information on food packages. The fridge communicates with the Ambient Assisted Living (AAL) system in the house and integrates itself. In particular, it detects the presence of spoiled food and discovers and receives a diet plan to be monitored based on what food items Mary is consuming.
An important part of Mary's diet is to ensure minimum liquid intake. The intelligent fridge partially contributes to it. To improve the accuracy, special sensor-enabled cups are used: some have sensors that beep when fluid intake is necessary and have a level to monitor the fluid consumed; others additionally have a gyro detecting spillage. They seamlessly coordinate in order to estimate the amount of liquid taken: the latter informs the former about spillages so that it can update the water intake level. However, Mary sometimes uses the cup to water flowers. Sensors in the faucets and in the toilet also provide a means to monitor this measurement.

Fig. 15. AAL case study.

Relax Requirement:

The fridge *SHALL* detect and communicate information with *AS MANY* food packages *AS POSSIBLE*.

ENV: Food locations, food item information (type, calories), food state (spoiled and unspoiled)

MON: RFID readers, Cameras, Weight sensors

REL: RFID tags provide food locations and food information; Cameras provide food locations (Cameras provide images that can be analyzed to estimate food locations), Weight sensors provide food information (whether eaten or not)

Fig. 16. RELAX requirement example.

5.1.1. High level goal model

Fig. 17 shows the high level goal model of the AAL. From the AAL system problem statement, we have identified *Reliability [AAL system]* as a non-functional high level goal. In fact, one of the expected qualities of the system is to run reliably. This is very important for several reasons and particularly because frequent visits from a technician could be a factor of disturbance for *Mary* and unfeasible due to the large number of AAL houses across the world. The high level goal *Reliability [AAL System]* is AND-refined into four sub-goals using refinement by type: *Precision [AAL System]*, *Security [AAL System]*, *Robustness [AAL System]* and *Performance [AAL System]*. Each sub-goal can be further refined until the refinement stops and we reach an *Elementary Goal* which can then be assigned to a *Contribution Goal*. The sub-goal *Precision [AAL System]* is AND-refined into two sub-goals: *Precision [Location Detection]* and *Precision [Sensors]* using refinement by subject. The sub-goal *Precision [Sensors]* is then AND-refined into three *Elementary NFGs* using refinement by subject. The sub-goal *Precision [Location Detection]* can be satisfied by a *positive* and *direct* contribution by one of the following *Contribution Goals: combine data from multiple sensors, combine multiple features* and *use redundant features*. The *Contribution Goal combine data from multiple sensors*, contribute indirectly and negatively to the satisfaction of the sub-goal *Performance [AAL System]*.

5.1.2. Low level goal model

Fig. 18 shows the security goal model of AAL. In order to further extract new goals from the AAL system, we identify another goal, *Security [fridge data]*, which is an *Abstract NFG* that can be AND-refined into three sub-goals using refinement by type: *Confidentiality [fridge data]*, *Integrity [fridge data]* and *Availability [fridge data]*. Similarly, the sub-goal *Availability [fridge data]* can be refined into two sub-goals using refinement by subject: *Availability [Storing RFID information]* and *Availability [Sensors data]*. The *Contribution Goal having high-end sensors* contributes directly and positively to the goal *Availability [Sensors data]*, and may contribute indirectly and positively to *Integrity [fridge data]*.

5.2. Properties verification of the AAL system with OMEGA2/IFx profile and toolset

The specification and verification of NFRs in the early stages of the AAL development cycle is a crucial issue (Nehmer et al., 2006). In this section, we show how we used OMEGA2/IFx (Verimag and Irit, 2011) for the properties verification and model simulation of AAL system.

5.2.1. Modeling the AAL system with OMEGA2 profile

We start by taking into account the structural part of the AAL system. Those parts are considered that are concerned with the daily calorie intake of the *Patient* in the AAL house. The AAL system is composed of *Fridge* and *Patient*; these parts are modeled along with the interaction that takes place between them. The *Fridge* partially contributes to the minimum liquid intake of the *Patient*; it also looks at the calorie consumption of the *Patient* as the *Patient* needs not to exceed it after a certain threshold.

Fig. 19 shows the main Internal Block Diagram (IBD). The communication between different blocks takes place through ports. In

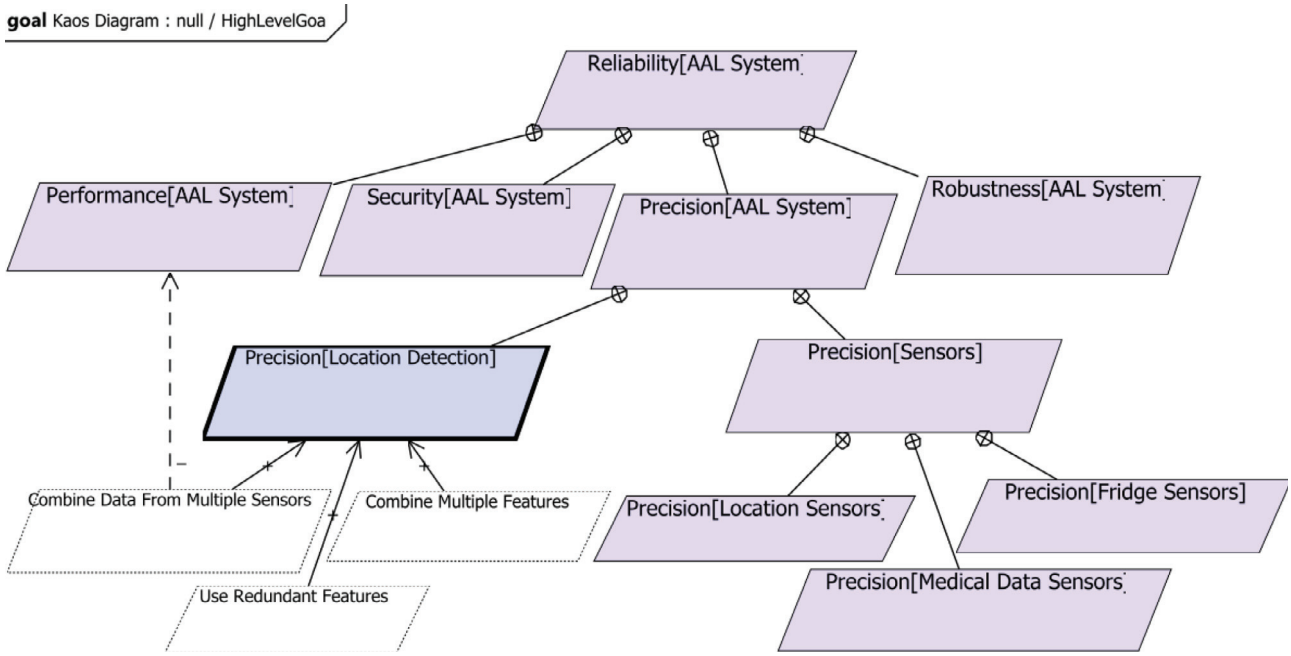


Fig. 17. High level goal model.

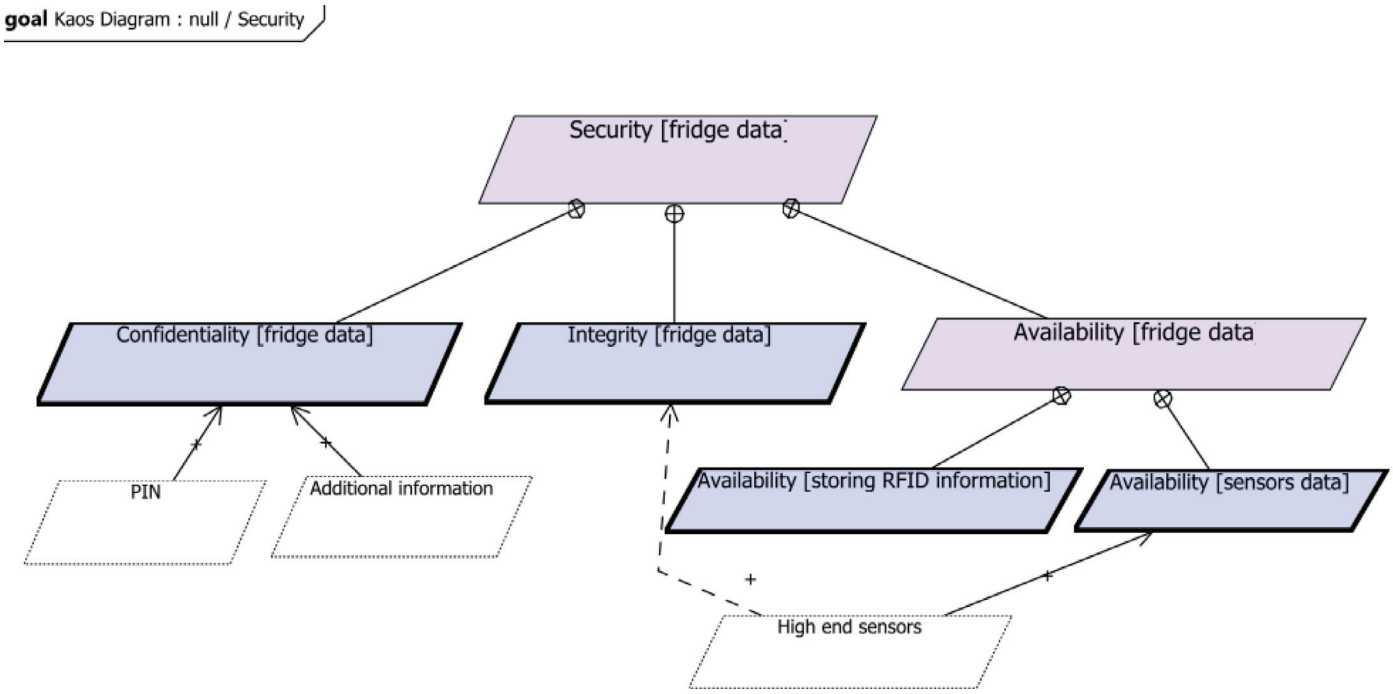


Fig. 18. Security goal model.

Fig. 19, the *Patient* block has a standard port named *pToFridge*. This port has a contract named *Patient2Fridge* and is acting as a provided interface of the *Patient* block. The important parts of the AAL system are *Patient* and *Fridge*. A *Fridge* in turn is composed of *Display, Alarm, Controller*, and *Food* blocks. Fig. 20 shows the IBD for the *Fridge* block. Each of the four blocks behaviors is modeled in a separate State Machine Diagram (SMD). The *Food* block contains information about the *Food* items in the *Fridge*, the calories contained in each item, the total number of calories the *Patient* has accumulated and the calorie threshold that should not be surpassed. The *Fridge Display* is used to show the amount of calories consumed by the *Patient*. The *Alarm* is activated in case the *Patient* calorie level surpasses a certain threshold.

Fig. 21 shows the SMDfor the *Patient* block. Here, the exchange of information between *Patient* and *Fridge* takes place. The number and quantity of each item present in the *Fridge* is identified. If a certain product still present in the *Fridge* is chosen by the *Patient* then the information is communicated with the *Fridge* and the list is updated. Otherwise the *Fridge* is empty and the *Patient* will wait to be refilled. Also, if the *Alarm* of the *Fridge* is raised due to high intake of calories, the *Patient* stops eating and waits for the system to be unblocked.

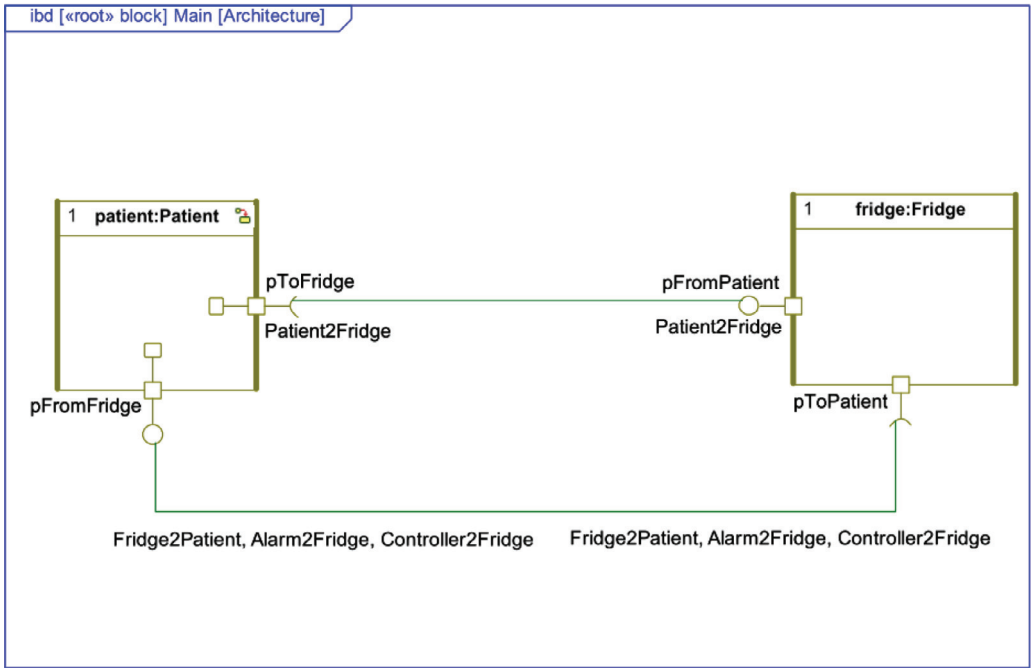


Fig. 19. Main Internal Block Diagram.

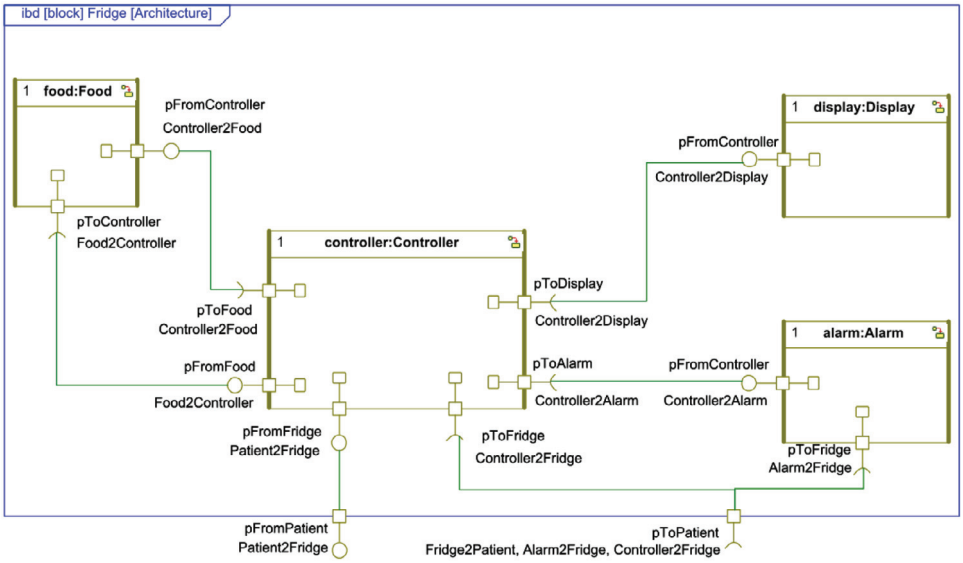


Fig. 20. Fridge Internal Block Diagram.

5.2.2. Properties verification of the AAL system

Below are the properties to be verified (Ahmad et al., 2013b).

Property 1: The Fridge SHALL detect and communicate information with AS MANY Food packages AS POSSIBLE. A RELAX-ed version of this requirement with all the uncertainty factors is shown in Fig. 16.

The satisfaction of this requirement contributes to the balanced diet of the *Patient*. The choice of this property for verification is motivated by the fact that it is important for the AAL system to know about as many *Food* items present in the *Fridge* as possible. Fig. 22 shows the SMD of the *Property 1*. The trigger for this property is an *observer* transition which is a *match* clause specifying the type of event (e.g., send), some related information (e.g., eat) and *observer* variable (e.g., p) that may send related information. The first task is to identify the number of items consumed by the *Patient* and the total number of items in the *Fridge*. Then the identity of the *Patient* is verified, if the person is identified as the *Patient*, then the next step is to calculate the number of items consumed. After this, the number of items left in the *Fridge* is calculated which is equal to the sum of all the items present in the *Fridge*. Then in the last step, we calculate if ((total number of items - number of items consumed - number of items left) >-1) and ((total number of items - number of items consumed - number of items left) <1), it means that we have reached the $<<success>>$ state by having information about all the items present in the *Fridge*, i.e., it should be 0 (which means that there is no information loss). Inversely, if it is less than or equal to -1 or greater than or equal to 1, then it means that we are missing information about some of the items present in the *Fridge* and the *observer* passes into the $<<error>>$ state.

We now consider the invariant requirement. *Property 2: The Alarm SHALL be raised instantaneously if the total number of calories surpasses the maximum calories allowed for the Patient.*Fig. 23 shows the SMDfor

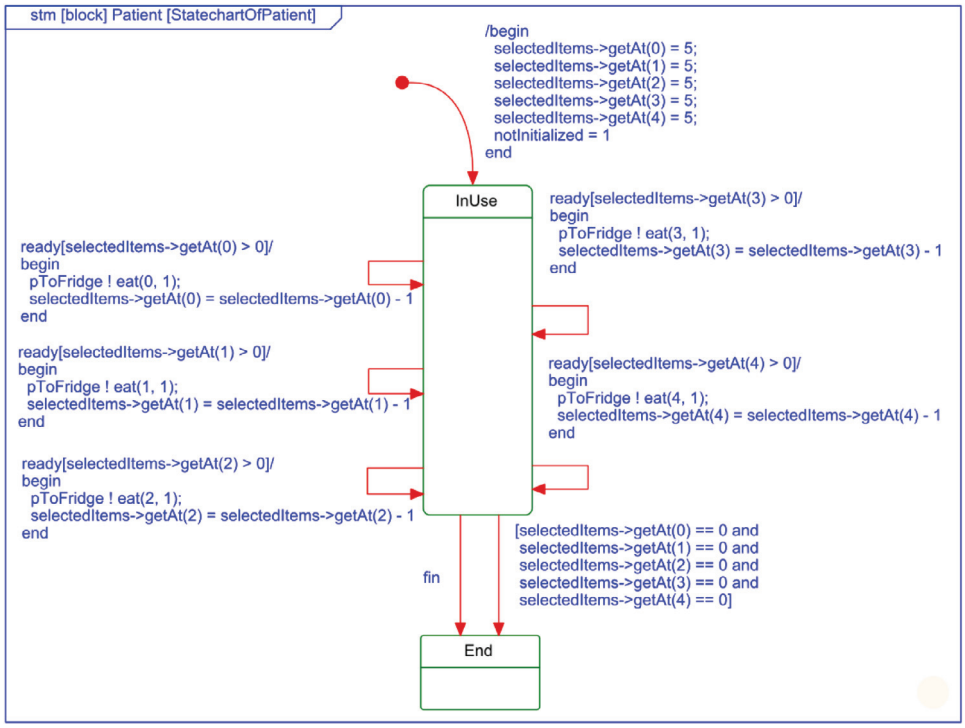


Fig. 21. Patient State Machine Diagram.

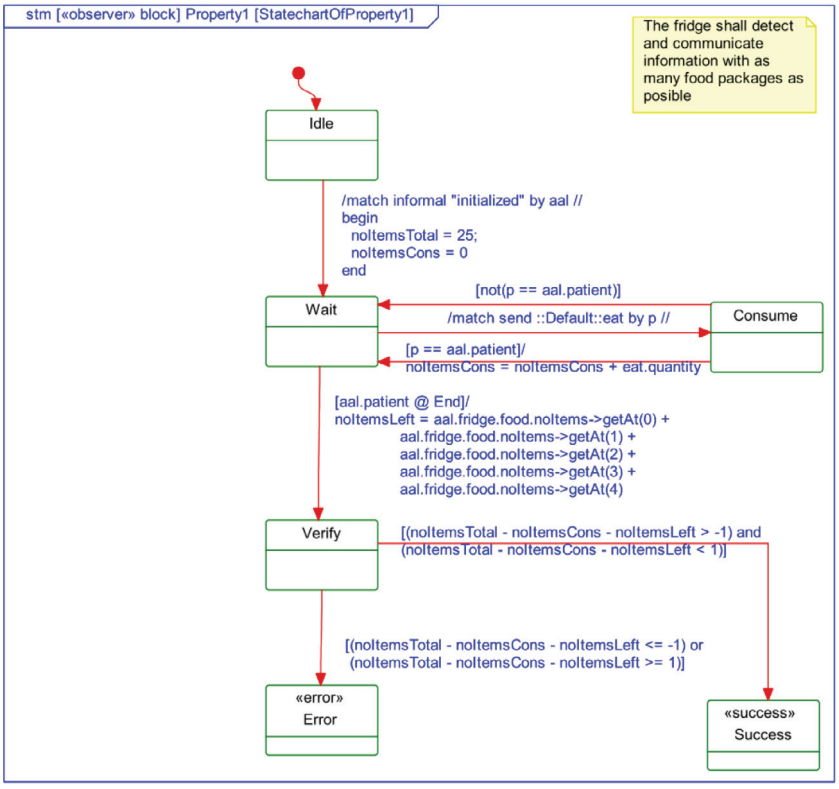


Fig. 22. Property1 State Machine Diagram.

property 2. This property ensures that the *Patient* should stop eating as soon as the total number of calories surpasses the maximum calories allowed and that the *Alarm* should be raised. This requirement implies that the *Alarm* shall be immediately raised as soon as the total number of calories equals or surpasses the maximum calories allowed for the *Patient*. If it happens then the *Patient* should stop eating and we will reach a $<<success>>$ state but if the *Patient* continues to eat, it means that we are reaching an $<<error>>$ state.

5.2.3. Verification results

Until now, the AAL system is modeled along with the properties to be verified on the model. We now show how to verify these

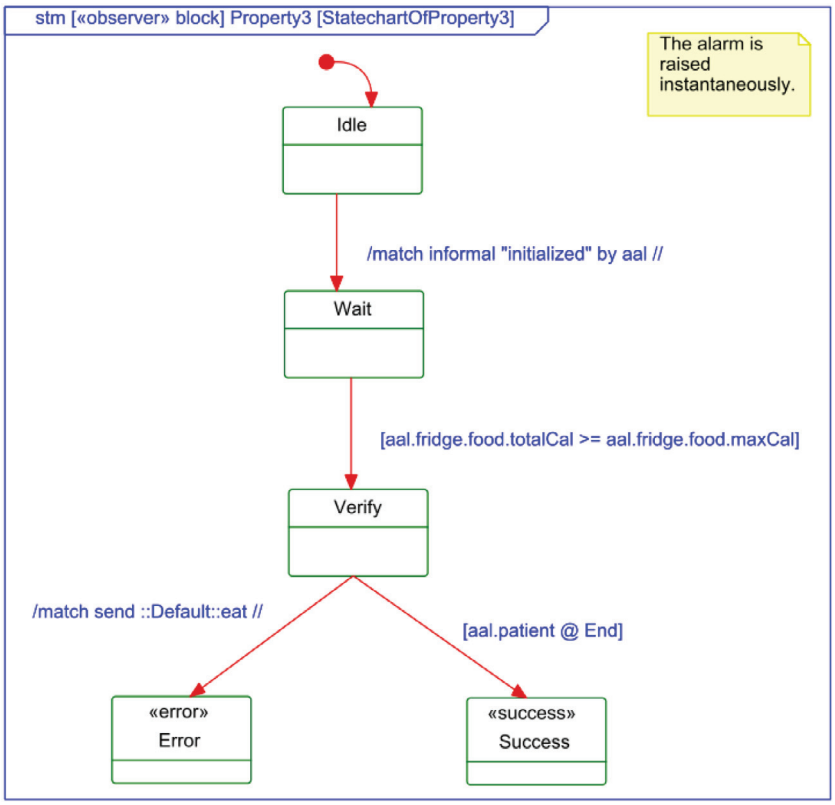


Fig. 23. Property2 State Machine Diagram.

properties using the IFx toolset. The AAL2 model is first exported into *AAL2.xmi* and then using the IFx toolset the *AAL2.xmi* is compiled into *AAL2.if* (Fig. 25). The *AAL2.if* is compiled into an executable file i.e., *AAL2.x* (Fig. 26). While verifying the AAL model, the model checker has found several error scenarios (Fig. 27). Any of the error scenarios can then be loaded through the interactive simulation interface of the IFx toolset to trace back the error in the model and then correct it.

In order to debug a model, firstly we import it into the simulator. We check the states of the *observers* in order to identify which property has not been satisfied. In this case, *Property 2* fails. While checking the state of the entire system for this property, we discover that the *<<error>>* state contained the maximum allowed number of calories for the total number of calories consumed and subsequently eat requests are sent by the *Patient*. This implies that the *Alarm* function of the intelligent *Fridge* does not function properly which is strictly linked to its *Food* process. One can observe in the SMDof the *Food* block (Fig. 24) that the *Alarm* is raised only if the total number of consumed calories is strictly superior to the maximum allowed; a condition which does not satisfy the request that the *Alarm* is raised as soon as possible. The correction consists of raising the *Alarm* also in case the total number of consumed calories is equal to the maximum allowed threshold. Once this error is corrected in the SMDof the *Food* block, the verification succeeds.

6. Conclusion and future work

The context of this research work is situated in the field of SE for SAS. This work resides in the very early stages of the software development life cycle i.e., at the RE phase. The overall contribution is to propose an integrated approach for modeling and verifying the requirements of SAS using Model Driven Engineering (MDE) techniques. It takes requirements as input and then by applying various processes and tools, we integrate the notion of uncertainty in requirements which we model using GORE techniques. Once we have the system design, we then introduce a mechanism for the properties verification of SAS.

We used Relax which is an RE language for SAS and which can introduce flexibility in NFRs to adapt to any changing environmental conditions. The essence of Relax for SAS is that it provides a way to relax certain requirements against other requirements in situations where the resources are constrained or priority must be given to requirements. For this purpose we have developed a tool called Relax COOL editor which is used to automate the formalization of SAS requirements by taking into account the different uncertainty factors associated with each Relax-ed requirement. We then use SysML/Kaos which is an extension of the SysML requirements model with concepts of the Kaos goal model. Here, invariant requirements are captured by the concept of FGs whereas Relax-ed requirements are captured by the concept of NFGs. We have provided a correlation table that helps in mapping the Relax and SysML/Kaos concepts. Using this table, the Relax-ed requirements are then transformed into SysML/Kaos goal concepts. This mapping is done using ATL, which is a model transformation technique and which takes as input a source model and transforms it into a target model. We have developed a tool called Relax2SysML/Kaos editor which is capable of modeling the Relax-ed requirements in the form SysML/Kaos goal concepts. We provide a mechanism to verify some adaptable and invariant properties of the SAS using formal method technique OMEGA2/IFx. In order to validate our proposed approach, we have applied it to an academic Ambient Assisted Living case study.

Our work resides within the framework of self-adaptation, but we do not treat the development of self-adaptation mechanisms. We help SAS developers by providing a mechanism for identifying the uncertainty associated with the requirements of these systems. Fig. 28 shows a table with the pros and cons of our proposed approach.

In terms of the future work, we have applied our approach to an academic case study. The next step is to apply it to a real industrial

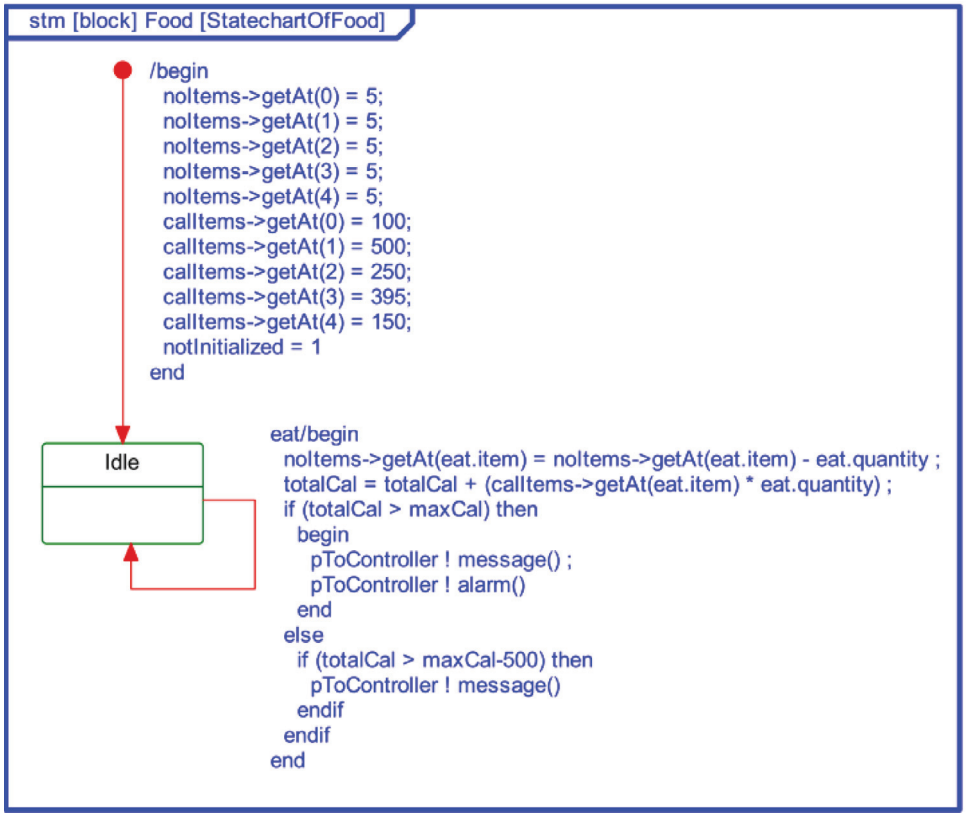


Fig. 24. Food State Machine Diagram.

```
[ahmad@topcased ~]$ uml2if -sysml -rhapsody -rhplang -eager AAL2.xmi
uml2if (OMEGA2) v2.0.1  (c) Verimag,IRIT 2009-2011
Analyzing input.
Please wait...
Success
Generated...
Preprocessed...
Indented...
Done.
[ahmad@topcased ~]$
```

Fig. 25. XMI to IF compilation.

```
[ahmad@topcased ~]$ if2gen -tdbm AAL2.if
Compiled IF spec...
m4 -I/home/dragomir/if2/src/code -D_CTYPE_=h AAL2.m4 > AAL2.h
m4 -I/home/dragomir/if2/src/code -D_CTYPE_=C AAL2.m4 > AAL2.C
g++ -c -o AAL2.o -Wall -Wno-deprecated -O3 -DABSTRACT -I/home/dragomir/if2/src/simulator-t -I/home/dragomir/if2/src/simulator AAL2.C
AAL2.C: In member function 'void if_Default_Property1_instance::_Idle_1_fire(IfMessage*)':
AAL2.C:24425: warning: deprecated conversion from string constant to 'char*'
AAL2.C: In member function 'void if_Default_Property1_instance::_Idle_1a_fire(IfMessage*)':
AAL2.C:24442: warning: deprecated conversion from string constant to 'char*'
AAL2.C: In member function 'void if_Default_Property2_instance::_Idle_1_fire(IfMessage*)':
AAL2.C:26060: warning: deprecated conversion from string constant to 'char*'
AAL2.C: In member function 'void if_Default_Property2_instance::_Idle_1a_fire(IfMessage*)':
AAL2.C:26077: warning: deprecated conversion from string constant to 'char*'
AAL2.C: In member function 'void if_Default_Property3_instance::_Idle_1_fire(IfMessage*)':
AAL2.C:26926: warning: deprecated conversion from string constant to 'char*'
AAL2.C: In member function 'void if_Default_Property3_instance::_Idle_1a_fire(IfMessage*)':
AAL2.C:26943: warning: deprecated conversion from string constant to 'char*'
AAL2.C: In member function 'void if_u2i__assumptions_instance::_s_1_fire(IfMessage*)':
AAL2.C:27853: warning: deprecated conversion from string constant to 'char*'
AAL2.C: In member function 'void if_u2i__assumptions_instance::_s_1a_fire(IfMessage*)':
AAL2.C:27871: warning: deprecated conversion from string constant to 'char*'
AAL2.C: In member function 'void if_u2i__assertions_instance::_ne_1_fire(IfMessage*)':
AAL2.C:28120: warning: deprecated conversion from string constant to 'char*'
AAL2.C: In member function 'void if_u2i__assertions_instance::_ne_1a_fire(IfMessage*)':
AAL2.C:28137: warning: deprecated conversion from string constant to 'char*'
Compiled C++ code...
g++ -o AAL2.x AAL2.o /home/dragomir/xrc/libxerces-c.so -L/home/dragomir/if2/bin/x86_64 -lsimulator -lexplorator
Done.
[ahmad@topcased ~]$
```

Fig. 26. IF to executable file compilation.

```
[ahmad@topcased ~]$ ./AAL2.x -dfs -po -me -ce 1n
00:11:34    2140556/s    5468814/t   326/d reached error state [2141463]
reached error state [2141465]
reached error state [2141467]
reached error state [2141474]
reached error state [2141476]
reached error state [2141478]
reached error state [2141488]
reached error state [2141490]
reached error state [2141492]
reached error state [2141515]
reached error state [2141517]
reached error state [2141519]
reached error state [2141532]
reached error state [2141534]
reached error state [2141536]
00:11:35    2143450/s    5472991/t   554/d reached error state [2143590]
```

Fig. 27. Model checker results in error scenarios.

	Pros & Cons of our Approach	Reasons
+	Proof of concepts	The proposed approach is validated on two case studies
+	Tools	Tools were developed to Implement the proposed approach
+	Usability of our Approach	Easily usable and understandable as our proposition is based on concepts already present in Requirements Engineering
-	Lack of Empirical Studies	As this research work was not part of an industrial project so we did not have the occasion to do some true empirical studies, we need to apply it on a real world case study to show the correctness of the transformation rules between Relax and SysML/Kaos concepts
-	No Demonstration of ROI	We provide no information regarding the Return On Investment by using our proposed approach
-	No adaptation Mechanism	We take into account the uncertainty in requirements of Self Adaptive Systems but we do not talk about the underlying adaptation mechanisms although we mention some techniques in the state of the art section to compare it with RELAX

Fig. 28. Pros and cons of our proposed approach.

case study, which will confront it to more rigorous and varied evaluation criteria such as its usability and its performance.

In order to validate the Relax-ed requirement, we check whether the new expression of the property is acceptable or not. If it is acceptable then we proceed with the next step, in case if it is not acceptable, we propose two options: i.e., to cancel the Relax-ation and go back to a *SHALL* invariant or complement the Relax-ed property with an additional invariant (e.g., a *max* or *min* boundary that constraints the Relax-ed expression). We would like to explore the validation step of the Relax-ed requirement in more detail, so that to show how we can introduce the boundary values in the Relax-ed expression.

We plan to investigate the adaptation mechanism techniques so that we can incorporate it in our proposed approach. Our approach takes into account the uncertainty in requirements of SAS, we model it using SysML/Kaos and then we verify it but we do not talk about the underlying adaptation mechanisms.

For the time being, our Relax2SysML/Kaos tool is capable of mapping the Relax concepts to SysML/Kaos concepts but not the inverse. A natural follow up of our work is to investigate how we could make it a two-way process, so that those people who are familiar with SysML/Kaos can map goal concepts to Relax concepts to which they are unfamiliar, so that to provide an additional knowledge base regarding requirements modeling of SAS. This would help in taking into account the information modeled in SysML/Kaos that we cannot capture in Relax.

The verification of Relax-ed requirements in our proposed approach is done using OMEGA2/IFx. To take into account the complexity of large systems, we can do the validation of their requirements at execution time. A promising approach to managing complexity in run-time environments is to develop adaptation mechanisms that leverage software models, referred to as Models@run.time (Blair et al., 2009). Research on Models@run.time seeks to extend the applicability of models and abstractions to the run-time environment, with the goal of providing effective technologies for managing the complexity of evolving software behavior while it is executing (Aßmann et al., 2011).

In our proposed approach, for the properties verification using OMEGA2/IFx, we model the *observers* and then we check these observers against the system design to see if the properties are verified or not. Right now, we model these *observers* as an SMD. We would like to automate this process of *observers* modeling by automatically generating it from Relax-ed and invariant requirements.

The use of model checking techniques used by OMEGA2/IFx exposes us to the problem of state space explosion which is inherent in these techniques. We handle this problem in our proposed approach by only injecting Relax-ed or invariant requirements, i.e., those requirements that are of interest for SAS. But we hope to tackle this problem using formal methods like B. There are already some works done for the mapping between SysML/Kaos and B in this regard. In Laleau et al. (2010), a method is defined for bridging the gap between the requirements analysis level (Extended SysML) and the formal specification level (B). This method derives the architecture of B specifications from SysML goal hierarchies. We believe that using proof-based formal methods like B can help in overcoming the state space explosion problem associated with model-checking techniques.

References

Abrial, Jean R., 1996. The B-Book: Assigning Programs to Meanings. Cambridge University Press, New York, NY, USA.

Ahmad, Manzoor, 2010. First step towards a domain specific language for self-adaptive systems. In: 10th Annual International Conference on New Technologies of Distributed Systems (NOTERE'10). IEEE, pp. 285–290.

Ahmad, Manzoor, 2013. Modeling and Verification of Functional and Non Functional Requirements of Ambient, Self Adaptive Systems (Ph.D. thesis). Mathématique Informatique Télécommunications, University of Toulouse Mirail, France.

Ahmad, Manzoor, Araújo, João, Belloir, Nicolas, Laleau, Régine, Bruel, Jean-Michel, Gnaho, Christophe, Semmak, FarridaRE) RE'13@, 2013a. Self-adaptive systems requirements modelling: Four related approaches comparison. In: Comparing ∗Requirements∗ Modeling Approaches Workshop (CMA@RE) RE'13. IEEE Computer Society Press, Rio de Janeiro Brazil, pp. 37–42.

Ahmad, Manzoor, Bruel, Jean-Michel, 2014. A comparative study of RELAX and SysML/Kaos. Technical Report. Institut de Recherche en Informatique de Toulouse, University Toulouse II Le Mirail, France.

Ahmad, Manzoor, Bruel, Jean-Michel, Laleau, Régine, Gnaho, Christophe, 2012a. Modélisation des Exigences pour les Systmes Auto-adaptatifs: Intégration des Techniques Relax/SysML/Kaos. In: Journées GDR - GPL - CIEL http://gpl2012.irisa.fr/sites/default/files/CIEL2012-Ahmad-paper22/index.pdf.

Ahmad, Manzoor, Bruel, Jean-Michel, Laleau, Régine, Gnaho, Christophe, 2012b. Using RELAX, SysML and KAOS for ambient systems requirements modeling. In: The 3rd International Conference on Ambient Systems, Networks and Technologies (ANT'12). Elsevier Procedia Computer Science, pp. 474–481.

Ahmad, Manzoor, Dragomir, Iulia, Bruel, Jean-Michel, Ober, Iulian, Belloir, Nicolas, 2013b. Early analysis of ambient systems sysml properties using omega2-ifx. In: 3rd International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH'13) SciTePress.

Apvrille, Ludovic, Courtiat, Jean P., Lohr, Christophe, de Saqui-Sannes, Pierre, 2004. TURTLE: A real-time UML profile supported by a formal validation toolkit. IEEE Trans. Softw. Eng. 30 (7), 473–487.

Aßmann, Uwe, Bencomo, Nelly, Cheng, Betty H.C., France, Robert B.Run.Time (Dagstuhl Seminar 11481)@, 2011. Models@Run.Time (Dagstuhl Seminar 11481). DagstuhlReports 1 (11) http://www.dagstuhl.de/de/programm/kalender/semhp/?semnr=11481.

Baresi, Luciano, Pasquale, Liliana, Spoletini, Paola, 2010. Fuzzy goals for requirements-driven adaptation. In: Proceedings of the 2010 18th IEEE International Requirements Engineering Conference. IEEE Computer Society, Washington, DC, USA, pp. 125–134.

Bascans, Jérémy, Walczak, Jérémy, Zeghoudi, Jérôme, Ahmad, Manzoor, Geisel, Jacob, Bruel, Jean-Michel, 2013. COOL RELAX Editor, M2ICE Project, Université de Toulouse le Mirail.

Benghazi, Kawtar, Visitación Hurtado, María, Rodríguez, María Luisa, Noguera, Manuel, 2009. Applying formal verification techniques to ambient assisted living systems. In: OnTheMove Workshop (OTM '09). Springer-Verlag, Berlin/Heidelberg, pp. 381–390.

Blair, Gordon S., Bencomo, Nelly, France, Robert B. Run.Time@, 2009. Models@Run.Time. Computer 42 (10), 22–27.

Bozga, Marius, Graf, Susanne, Ober, Ileana, Ober, Iulian, Sifakis, Joseph, 2004. The IF toolset. In: Formal Methods for the Design of Real-Time Systems (FMDRTS '04). Springer-Verlag, Berlin/Heidelberg, pp. 237–267.

Cheng, Betty H.C., Sawyer, Pete, Bencomo, Nelly, Whittle, Jon, 2009a. A goal-based modeling approach to develop requirements of an adaptive system with environmental uncertainty. In: Proceedings of the 12th International Conference on Model Driven Engineering Languages and Systems (MODELS'09). Springer-Verlag, Berlin/Heidelberg, pp. 468–483.

Cheng, Betty H.C., de Lemos, Rogério, Giese, Holger, Inverardi, Paola, Magee, Jeff, Andersson, Jesper, et al., 2009b. Software engineering for self-adaptive systems: A research roadmap. In: Software Engineering for Self-Adaptive Systems. Springer-Verlag, Berlin, Heidelberg, pp. 1–26.

Chung, Lawrence, Nixon, Brian A., Yu, Eric, Mylopoulos, John, 1999. Non-Functional Requirements in Software Engineering, 1st Springer-Verlag.

Clarke, Edmund M., Grumberg, Orna, Peled, Doron, 1999. Model Checking. MIT Press, London.

Clarke, Edmund M., Klieber, William, Novek, Milo, Zuliani, Paolo, 2012. Model checking and the state explosion problem. In: Meyer, Bertrand, Nordio, Martin (Eds.), Tools for Practical Software Verification. In: Lecture Notes in Computer Science, 7682. Springer-Verlag, Berlin Heidelberg, pp. 1–30.

Cleland-Huang, Jane, Settimi, Raffaella, Zou, Xuchang, Solc, Peter, 2007. Automated classification of non-functional requirements. Requir. Eng. 12 (2), 103–120.

Courtiat, Jean P., Santos, Celso A.S., Lohr, Christophe, Outtaj, B., 2000. Experience with RT-LOTOS, a temporal extension of the LOTOS formal description technique. Comput. Commun. 23 (12), 1104–1123.

Cysneiros, Luiz Marcio, Leite, Julio Cesar Sampaio do Prado, 2004. Non functional requirements: From elicitation to conceptual models. IEEE Trans. Softw. Eng. 30, 328–350.

de Lemos, Rogério, Giese, Holger, Müller, A. Hausi, Shaw, Mary, Andersson, Jesper, Litoiu, Marin, et al., 2013. Software engineering for self-adaptive systems: A second research roadmap. In: de Lemos, R., Giese, H., Mller, H., Shaw, M. (Eds.), Software Engineering for Self-Adaptive Systems II. In: Lecture Notes in Computer Science, 7475. Springer-Verlag, Berlin Heidelberg, pp. 1–32. doi:10.1007/978-3-642-35813-5_1.

Dragomir, Iulia, Ober, Iulian, Lesens, David, 2012. A case study in formal system engineering with SysML. In: 17th International Conference on Engineering of Complex Computer Systems (ICECCS'12). IEEE, pp. 189–198.

Gnaho, Christophe, Semmak, Farida, 2010. Une Extension SysML pour l'ingénierie des Exigences Non-Fonctionnelles Orientée But. In: Ingénierie des Systèmes d'Information. Lavoisier Paris FRANCE, pp. 277–292.

Goldsby, Heather J., Sawyer, Pete, Bencomo, Nelly, Cheng, Betty H.C., Hughes, Danny, 2008. Goal-based modeling of dynamically adaptive system requirements. In: Proceedings of the 15th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems. IEEE Computer Society, Washington, DC, USA, pp. 36–45.

Kasten, Eric P., Sadjadi, Seyed M., McKinley, Philip K., 2003. Architecture and operation of an adaptable communication substrate. In: Proceedings of the Ninth IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS'03) IEEE http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1204293&queryText=Architecture+and+operation+of+an+adaptable+communication+substrate&newsearch=true&searchField=Search_All.

Kephart, Jeffrey O., Chess, David M., 2003. The Vision of Autonomic Computing. Computer 36 (1).

Laleau, Régine, Semmak, Farida, Matoussi, Abderrahman, Petit, Dorian, Hammad, Ahmed, Tatibouet, Bruno, 2010. A First Attempt to Combine SysML Requirements Diagrams and B. Innovations in Systems and Software Engineering 6.

Lamsweerde, Axel V., 2009. Requirements Engineering: From System Goals to UML Models to Software Specifications, 1st edition Wiley.

Lapouchnian, Alexei, Liaskos, Sotirios, Mylopoulos, John, Yu, Yijun, 2005. Towards Requirements-Driven Autonomic Systems Design. In: Proceedings of the 2005 workshop on Design and evolution of autonomic application software. ACM, New York, NY, USA, pp. 1–7.

Lutz, Robyn R., 1993. Targeting safety-related errors during software requirements analysis. J. Syst. Softw. 34 (3), 223–230.

Moon, Seong ick, Lee, K.H., Lee, Doheon, 2004. Fuzzy branching temporal logic. IEEE Trans. Syst. Man Cybernet. B: Cybernet. 34 (2).

Nehmer, Jürgen, Becker, Martin, Karshmer, Arthur, Lamm, Rosemarie, 2006. Living assistance systems: An ambient intelligence approach. In: Proceedings of the 28th International Conference on Software Engineering (ICSE'06). ACM, pp. 43–50.

Ober, Iulian, Dragomir, Iulia, 2010. OMEGA2: A new version of the profile and the tools. In: 15th International Conference on Engineering of Complex Computer Systems (ICECCS '10). IEEE, pp. 373–378.

Ramirez, Andres J., Cheng, Betty H.C., Bencomo, Nelly, Sawyer, Pete, 2012a. Relaxing claims: Coping with uncertainty while evaluating assumptions at run time. In: France, Robert B., Kazmeier, Jrgen, Breu, Ruth, Atkinson, Colin (Eds.), Model Driven Engineering Languages and Systems. In: Lecture Notes in Computer Science, 7590. Springer-Verlag, Berlin/Heidelberg, pp. 53–69.

Ramirez, Andres J., Fredericks, Erik M., Jensen, Adam C., Cheng, Betty H.C., 2012b. Automatically RELAXing a goal model to cope with uncertainty. In: Proceedings of the 4th International conference on Search Based Software Engineering. Springer-Verlag, Berlin/Heidelberg, pp. 198–212.

Verimag, Irit, 2011. OMEGA2-IFx for UML/SysML v2.0, Profile and Toolset, User Manual Document v1.1.

Vítor, E., Souza, S., Lapouchnian, Alexei, Robinson, William N., Mylopoulos, John, 2011. Awareness requirements for adaptive systems. In: Proceedings of the 6th International Symposium on Software Engineering for Adaptive and Self-Managing Systems. ACM, New York, NY, USA, pp. 60–69.

Welsh, Kristopher, Sawyer, Pete, 2010. Understanding the scope of uncertainty in dynamically adaptive systems. In: Requirements Engineering: Foundation for Software Quality. Springer-Verlag, Berlin Heidelberg, pp. 2–16.

Welsh, Kristopher, Sawyer, Pete, Bencomo, Nelly, 2011. Towards requirements aware systems: Run-time resolution of design-time assumptions. In: Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering. IEEE Computer Society, pp. 560–563.

Whittle, Jon, Sawyer, Pete, Bencomo, Nelly, Cheng, Betty H.C., 2008. A language for self-adaptive system requirements. In: International Workshop on Service-Oriented Computing: Consequences for Engineering Requirements (SOCCER'08) IEEE http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4797489&filter%3DAND%28p_IS_Number%3A4797485%29.

Whittle, Jon, Sawyer, Pete, Bencomo, Nelly, Cheng, Betty H.C., Bruel, J.-M., 2009. RELAX: Incorporating uncertainty into the specification of self-adaptive systems. In: Proceedings of the 2009 17th IEEE International Requirements Engineering Conference, RE. IEEE Computer Society, Washington, DC, USA, pp. 79–88.

Yu, Eric S.K., 1997. Towards modeling and reasoning support for early-phase requirements engineering. In: Proceedings of the 3rd IEEE International Symposium on Requirements Engineering. IEEE Computer Society, pp. 226–235.

Yu, Yijun, Lapouchnian, Alexei, Liaskos, Sotirios, Mylopoulos, John, Leite, Julio C.S.P., 2008. From goals to high-variability software design. In: Proceedings of the 17th International Conference on Foundations of Intelligent Systems. Springer-Verlag, Berlin, Heidelberg, pp. 1–16.

Yu, Yijun, Leite, Julio C.S.P., Mylopoulos, John, 2004. From goals to aspects: Discovering Aspects from requirements goal models. In: Proceedings of the 12th IEEE International Requirements Engineering Conference. IEEE Computer Society, Washington, DC, USA, pp. 38–47.

Manzoor Ahmad received his Ph.D. from the University of Toulouse Mirail in October 2013. He is working as research/teacher assistant at the University of Pau for the academic year 2013-2014. Currently member of the LIUPPA (Laboratoire d'Informatique de l'Université de Pau et des Pays de l'Adour). Previous Member of the MACAO team (Modèles, Aspects, Composants pour des Architectures à Objets) of the IRIT (Institut de Recherche en Informatique de Toulouse) CNRS laboratory. His research areas include requirements engineering, model driven engineering, goal based requirements engineering, computer networks and use of formal methods for the properties verification of self-adaptive systems.

Nicolas Belloir is an associate professor at the computer department of the University of Pau, France, since 2005. It is member of the MOVIES research team at the LIUPPA (Laboratoire d'Informatique de l'Université de Pau et des Pays de l'Adour). He received a Ms. D. in computer science from the University of Toulouse in 1999. He worked as software engineer between 1999 and 2001 in Transiciel Company on embedded and real-time systems. He received its Ph. D. in computer science from the University of Pau in 2004. His research interest deals with semi-formal modeling language (UML, SysML, DSML …), model driven engineering, requirement engineering, and component-based software engineering.

Jean-Michel Bruel received his Ph.D. from the University Paul Sabatier (Toulouse) in December 1996. From September 1997 to August 2008, he was associate professor at the University of Pau. Member of the LIUPPA (Laboratoire d'Informatique de l'Université de Pau et des Pays de l'Adour) from 2000 to 2008. Currently member of the MACAO team (Modèles, Aspects, Composants pour des Architectures à Objets) of the IRIT (Institut de Recherche en Informatique de Toulouse) CNRS laboratory. His research areas include development of distributed, component-based applications, methods integration, and on the use of formal methods in the Component-Based Software Engineering context. He has defended his "Habilitation à Diriger des Recherches" in December 2006 and obtained in 2008 a full professor position at the University of Toulouse. He has been head of the Computer Science department of the Technical Institute of Blagnac from 2009 to 2012.

Received: 31 January 2019 | Revised: 14 October 2019 | Accepted: 16 October 2019

DOI: 10.1002/sys.21523

REGULAR PAPER

WILEY

Systems of systems: From mission definition to architecture description

Imane Cherfa[1,2,5] | Nicolas Belloir[2,3] | Salah Sadou[2] | Régis Fleurquin[2] | Djamal Bennouar[4]

[1]Sciences Faculty, CS Department, University of BLIDA1 - LRDSI, Algeria

[2]Université Bretagne Sud - IRISA, France

[3]CREC St-Cyr, Military Academy of St Cyr Coetquidan, France

[4]University of Bouira - LIMPAF, Algeria

[5]Present address: Nicolas Belloir, IRISA, Université de Bretagne-Sud, Campus de Tohannic, bâtiment ENSIBS, Rue Yves Mainguy, BP 573, 56017 Vannes cedex, France

Correspondence
Nicolas Belloir, IRISA, Université de Bretagne-Sud, Campus de Tohannic, Bâtiment ENSIBS, Rue Yves Mainguy, BP 573, 56017 Vannes Cedex, France.
Email: nicolas.belloir@irisa.fr

Abstract

Systems of Systems (SoS) encompass a group of distributed and independent systems. This class of systems requires recurrent adaptation at runtime owing to the uncertainty and variability of the runtime environment. Thus, during their execution, SoS can deviate from the initial specification, which is often a consequence of successive evolutions. This problem occurs mainly due to (a) weak communication between the SoS analysis stage and architecture stage and (b) the lack of links between the operational planning in the SoS analysis stage and systems that must be involved in the SoS architecture stage. This paper proposes a model-based process that strengthens the links between the SoS analysis stage and the architecture stage in the wave life cycle. We ensure that the mission and role concepts for the SoS definition are sufficiently abstract to allow adaptation to the variability of the environment. This definition is translated into an abstract architecture that guides the choices of the system architect during the design and evolution stages. The proposed language is an adaptation of the Systems Modeling Language (SysML). Furthermore, we define a crowd management SoS to illustrate the process.

KEYWORDS

mission, model-based process, systems of systems

1 | INTRODUCTION

Model-based approaches[1] represent a promising path for the development and analysis of systems of systems (SoS). These approaches allow the control of the overall complexity of the SoS, clarification and documentation of its structure and behavior, and communication of these aspects to the stakeholders.[2] Models can be developed at different stages during the SoS development process, expressing different points of views. Consequently, choices and decisions can be made at each stage of the development process using different models. For instance, the application domain expert (ADE) makes decisions pertaining to business aspects, while the system architect makes decisions that lead to certain implementation choices. Thus, the first stakeholder is concerned with the need and the requirement stage, while the second one is concerned with the design stage. These two stages are crucial for system development as they form its base. Thus, the choices made during the design must be consistent with the decisions made during the definition of the requirements. However, the risk of loss of information is also the highest between these two stages,[3] as the stakeholders concerned with these two stages often come from considerably different domains, and different domains involve risks of misunderstanding.

This problem concerns only the consistency of the information transmitted from the requirement stage to the design stage. However, in the case of SoS, whose nature inherently involves evolution,[4] the consistency of the choices made for the evolution needs with the initial capability objectives must be verified. The evolutions of an SoS can occur far in time, which means that some or all of the initial stakeholders may no longer be present. In such a case, without strong links between the choices made during the design stage and the requirements defined in the previous stage, the risk that the SoS deviates from its original objectives is considerably high, which is undesirable.

To solve this problem, we propose the creation of a strong link between the SoS analysis stage and the architecture stage in the SoS wave life cycle that is dedicated to the acknowledged SoS. The main concept is to make the SoS analysis stage closer to the architecture stage. This aspect is achieved by using a language sufficiently familiar to the ADE to clearly articulate her/his needs and requirements and

sufficiently formal to serve as a guide and controller for the system architect during the design and evolution stages. We propose the use of our language in the mission paradigm[5–8] for the definition of the SoS. The goal is that the mission must be sufficiently defined to assist the architect to determine systems that must be involved and the functions that these systems must perform. However, in the case of the definition of SoS, which operate in dynamic environments, the ADE cannot always predict the constituent systems that will actually exist when the SoS is launched. Reasonably, we can consider that the ADE knows the "types" of systems that her/his SoS needs. Therefore, during the definition of the mission by using our language, we refer to roles (abstract entities) instead of concrete systems.

Once the mission is completely defined by the ADE, it is translated by the architect to an abstract architecture holding invariants that can guide the choices among the possible solutions during the design and evolution of the SoS. In other words, if the architect tries to use a solution that is inconsistent with one of the ADE requirements, she/he will be notified of this aspect with information pertaining to the related consequences. Thus, the abstract architecture is the solid bridge between the SoS analysis stage and the architecture stage.

The remaining paper is organized as follows. In Section 2, we present the background and motivation. Section 3 discusses the state of the art and presents a conceptual model that serves as a basis for the proposed approach. Section 4 illustrates our general approach for SoS development. Section 5 describes the mission modeling, details the proposed profile for describing the capabilities, and explains the architecture creation process. In Section 6, we illustrate the proposed approach through our case study, while Section 7 presents the concluding remarks and directions for future work.

2 | BACKGROUND AND MOTIVATION

As might be envisaged in an emerging field, there does not exist unifying definition of SoS. The several definitions of SoS have their own merits, depending on their application domain.[4,9,10] To bind the field precisely without considering any application domain, Maier[11] characterized SoS in terms of five principal features: "operational independence, managerial independence, geographic distribution, evolutionary development, and emergent behavior."

According to several authors,[11,12] the independence of the constituent systems is the main feature of SoS. Each constituent system assumes its own goals and operates independently.[13] The need to maintain autonomy while simultaneously operating within the SoS context considerably increases the complexity of an SoS and is at the heart of the SoS Systems Engineering (SE) challenge.[14] As discussed by Ref. 15, when using traditional SE approaches such as Object-Oriented Systems Engineering Method (OOSEM)[16,17] and Harmony-SE[18] systems engineers can trace system boundaries and define requirements clearly. Furthermore, the engineers can master the development environment to assure that the technical trade studies are the basis of the allocations of requirements to components. In the SoS environment, systems engineers must take into account considerations beyond the use of existing systems as the constituent systems of these SoS. They must allocate the realization details and functionalities, which may not be optimal from the point of view of the SoS. Furthermore, constituent systems must retain their independence. For these reasons, SE is highly challenging in the SoS context.

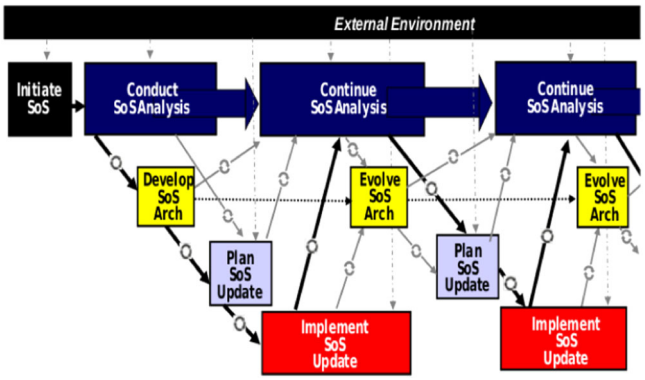


FIGURE 1 SoS SE wave model[12]

To guide the selection of SoS SE principles, the DoD[19] categorized SoS based on their degree of centrality into "virtual, collaborative, acknowledged, and directed SoS." Virtual SoS is characterized by the absence of central management and common goal. Collaborative SoS is characterized by the absence of a central authority, constituent systems of collaborative SoS interoperate more or less voluntarily to achieve the main common goals. The acknowledged SoS is under the responsibility of an organization, that sustains the SoS SE while the constituent systems preserve their independent development and goals. Directed SoS are developed and supervised to meet special purposes. Constituent systems can operate independently but are controlled to fulfill the SoS goal. SoS SE is primarily concerned with acknowledged SoS.

To address SoS SE challenges, the U.S. Department of Defense (DoD) published the SE guidance for SoS[19] and proposed the trapeze model. Seven core elements characterize the trapeze model and are described as follows[15,19]: (a) "translating the SoS capability objectives into requirements" and (b) "assessing the performance pertaining to these capability objectives" as well as (c) "monitoring and assessing the external changes on the SoS." It is important for SE for SoS to (d) "understand systems that contribute to the realization of SoS objectives and their relationships" and (e) "to develop and evolve an SoS architecture." For the SE for SoS, it is crucial to (f) "address new requirements and solution options" and (g) "orchestrate upgrades to the SoS and implement the changes."

Dahmann et al[12] "built on the trapeze model and translated the SoS SE core elements, their interrelationships, and SoS decision making artifacts to a more familiar and intuitive wave model representation." Originally, the wave planning was introduced by Dombkins,[20] and it was subsequently "applied to the SoS trapeze model to illustrate the incremental and iterative process that characterizes acknowledged SoS development."[12] The SoS SE wave model is illustrated in Figure 1.

Six steps characterize the SoS SE wave model.[12] In the SoS SE wave model, systems engineers are actors in (a) initiating the SoS by

understanding SoS goals and (b) conducting SoS analysis by taking into account several artifacts such as SoS performance measures and SoS risks and mitigations. Fundamental to SoS SE is (c) the development and evaluation of the SoS architecture as well as (d) the planning of SoS updates and evaluation of the SoS priorities. The SoS SE team is involved in (e) monitoring the implementations at the constituent system level and finally (f) performing a continual SoS analysis to revisit key information.

The wave model helps to improve the SoS SE design capabilities and approaches to manage the SoS problems. However, more recent research argues that in SoS SE, the design and the end to end process and management are required to be balanced.[5,6] In fact, SoS are acquired to satisfy new capabilities in a mission context. The latter is a key element to assist SoS engineers to determine the systems that must be involved and the functions they must perform.[5,6] It is important to consider the mission thread to bridge the dissociation between the SoS objectives and the individual functionalities undertaken by the systems that constitute the SoS to support the SoS mission. "The allocations of functions or activities to constituent systems can dynamically change over the course of a mission thread."[5] To address these SoS SE challenges, we propose in this paper the maintenance of a mission focus throughout the SoS SE analysis and the architecture process included in the wave model.

3 | STATE OF THE ART

Our work concerns three domains of state of the art: (a) SE, (b) SoS SE, and (c) mission engineering (ME). In what follows, we discuss the existing work related to these three areas.

3.1 | SE approaches

A number of SE methodologies are currently used by the SE community, offering guidance for analyzing, developing, and documenting complex systems. "The Object-Oriented Systems Engineering Method (OOSEM) provides an integrated framework that combines object-oriented techniques, a model-based design approach, and traditional top-down SE practices,"[17] OOSEM is now advocated as an example of a model-based systems engineering (MBSE) best practice since it was realigned with Systems Modeling Language (SysML) (it was initially based on the unified modeling language [UML]). The following activities are encompassed in the OOSEM "specification and design system process"[17]: (a) analysis of the stakeholder needs, and (b) analysis of the system requirements. (c) Logical architecture definition by highlighting how the logical components interact to fulfill the requirements. (d) The allocation of hardware, software, data, and procedures to the logical components. (e) The activity of optimizing and evaluating alternatives, and finally, (f) the activity of managing the traceability of requirements from the mission level requirements to the component requirements.

The Harmony[18,21] SE process is characterized by three main activities: (a) "requirements analysis," (b) "system functional analysis," and (c) "architectural design." The Harmony SE is a model-based process and uses SysML as the modeling language. In the Harmony "requirements analysis phase," requirements are grouped into use cases. The "system functional analysis" phase consists of translating functional requirements into set of system functions. A black box model is related to each use case. Incrementally, these black box models are aggregated into a black box system model. The "architectural design" activity is composed of the "system architectural design" and "subsystem architectural design" elements. In the subsequent system architectural design phase, the valid operational contracts are assigned, based on performance and safety requirements, to the physical architecture. The subsequent subsystem architectural design phase aims at deciding the operational contracts within a physical subsystem that should be implemented in the hardware and software (hardware/software trade-off analysis).

MagicGrid[22] is a SysML-based framework for modeling complex systems. The MagicGrid framework consists of viewpoints (black box, white box, and solution) and aspects (the four pillars of SysML: requirements, system structure, system behavior, and parameters) organized in a grid view. The cells of the grid represent different views of MBSE, which are described as follows[22]: (a) the requirement elicitation of stakeholders by using the SysML requirement (RE) diagram; (b) a use case description of the refinements of functional stakeholder needs; (c) system context representation using the SysML internal block diagram (IBD); (d) measures of effectiveness (MoEs), which indicate the nonfunctional requirements, described in the SysML block definition diagram (BDD). The MoEs calculation procedures are specified with the SysML parametric diagrams. (e) The identification and specification of system requirements is performed by using the RE diagram, and (f) functional analysis elaboration is performed with multiple SysML activity diagrams, specifying internal system functions. (g) The logical subsystem communication identification is established using the control and resource flows defined in the functional analysis model. Both of the SysML BDD and SysML IBD are used to capture this view. (h) The MoEs as well as the measures of performance (MoPs) are captured for each logical subsystem, in the SysML BDD and parametric diagrams. (i) The component requirements are captured using the SysML RE diagram. (j) The component behavior definition is performed using an association of SysML state machine, activity, and sequence diagrams; (k) component structure elaboration is performed by illustrating the physical connections between physical components, and this view is captured using both the SysML BDD and IBD. (l) The component parameter definition of each component is performed, in which each parameter captures the component characteristics and the links between them and describes how the MoEs and MoPs already specified are accomplished using these characteristics.

Other interesting SE approaches have also been reported in the literature, such as the IBM Rational Unified Process for Systems Engineering (RUP SE),[23] JPL State Analysis (SA),[24] and SYStem MODeling (SYSMOD). It is clear that the SE approaches are used to solve different tasks of the SE process,[17] and they have reached a level of maturity in the identification and gathering of artifacts and best practices for complex SE. However, SE approaches can trace system boundaries and define requirements clearly,[15] something that is not obvious in the SoS

SE. Furthermore, trade analyses and measures of performance allow the optimal allocation of components to requirements while in SoS SE, SoS engineers need to take into account considerations beyond the use of existing systems as the constituent systems of these SoS.[15] They must allocate the realization details and functionalities that may not be optimal from the SoS point of view. Moreover, SoS engineers do not control the overall development environment of the SoS because the constituent systems must retain their independence.

3.2 | SoS SE contributions

In terms of the SoS SE, the DoD has published the SE Guidance for SoS,[19] which provides a well-grounded practical guidance for systems of systems. According to the guide, seven main elements characterize the SoS SE (the trapeze model already described in Section 2), each element "can be mapped to the 16 technical management processes" defined in the Defense Acquisition Guidebook.[25]

Dahmann et al[12] built on the trapeze model and proposed a translation of the trapeze model elements into the wave model rationale. The main elements of the wave model have been presented in Section 2 and Figure 1. The critical information to realize effective SoS SE, and the corresponding artifacts were identified by Dahmann et al.[26] As we are interested only in the analysis and architecture phases of the wave model, in the following section, we present only the artifacts related to these two stages. The artifacts are described as follows[26]: (a) The SoS capability objectives correspond to the main goals of the SoS; (b) the SoS CONcept Of OPerationS (CONOPS) defines the use of the SoS constituent system functionality in an operational context; (c) the systems information corresponds to information pertaining to the systems that impacts the SoS capability objectives, and this information is collected to be used for replacements as the SoS evolves; (d) the SoS requirement space bounds the operational tasks and missions while considering the environment change that affect the execution of the required functions; (e) the SoS performance measures and methods capture the basis for assessing the overall performance of the SoS and for improving the SoS; (f) the effectiveness data of the SoS are collected from different environments to identify the areas needing more attention; (g) the SoS SE planning elements determine "the rhythm, technical reviews, and decision processes across the SoS evolution." These elements furnish also "the principal SE rules of engagement for the SoS and are utilized by all SoS actors"; (h) the SoS risks are captured and tracked.

The Department of Defense Architecture Framework (DoDAF)[27] and the Ministry of Defense Architecture Framework (MoDAF)[28] are, respectively, the architectural frameworks for the DoD and the UK Ministry of Defense (MoD). Both of DoDAF and MoDAF provide set of views, each of which is decomposed into products and data, for instance, operational view, capability view, and systems and services view. The operational view aims to describe the tasks and activities, operational elements, and resource flow exchanges required to conduct operations. The capability view aims to describe the mapping between the required capabilities and the activities that enable those capabilities. While the two frameworks have similar views, their respective metamodels are different. Despite this, the Object Management Group (OMG) proposed the UML Profile for DoDAF and MoDAF (UPDM)[29], a common modeling language for both frameworks that is based on the UML but can either be used with SysML. The UPDM prescribes more than 40 views. The viewpoints allow modeling in different levels of abstraction, and are rich in term of concepts, including all the concepts related to SE or SoS SE domains. However, the selection of a viewpoint can be difficult, and it is difficult to take advantage of the interconnected views because none of these views provide a simplified perspective that addresses only the subsets of each view. The architectural frameworks constitute "in depth modeling approach, which requires significant resources."[30]

Previous work, such as the DANSE project[31] and the Compass project[32] have proposed a reduction in the architecture frameworks according to the target objectives. DANSE proposed that one should focus on the selected views of the UPDM instead of considering all the views. The SoS mission is described using the Operational View OV-1. Subsequently, Operational View OV-5 specifies the tasks to be involved to achieve the SoS mission. The Operational View OV-2 is used to define the data exchange within the system. The functionalities that can implement the capabilities are determined in the System View SV-5. Finally, System View SV-10A expresses the functional and non-functional constraints. Compass proposed to delimit SoS boundaries in early stage, while SoS modeling process must taking into account that SoS environment is open.

3.3 | Mission engineering

Mission Engineering (ME) is an emerging field owing to the need for understanding and documenting the end to end mission execution in an SoS.[5,6] ME "*combines the structure of systems engineering and the tactical insights of operational planning to a system of systems to deliver a specific capability. The difficulty in ME revolves around the concept of a mission context, which manages the uncertainties, dynamics and stochastic behaviors of SoS.*"[5]

In terms of software intensive SoS, Silva et al[33] proposed M2Arch, a model-based refinement process for SoS architectural modeling that uses missions as the basis. The mission model is defined using mKAOS,[34] an SoS mission description language. In mKAOS, the mission is the specialization of a goal to the SoS domain. The mission is refined with and/or operators until sub-missions that can be handled by a constituent system are determined. The authors defined an SoS mission as encompassing five concepts: (a) priority, (b) trigger, (c) constraints, (d) parameters, and (e) tasks, which are functional operations to be executed. The software architecture is generated automatically in SosADL,[33] a formal language to describe SoS software architectures. However, this work did not consider the hardware and human constituent systems. Moreover, even if the mKAOS language allows the definition of an emergent behavior model, that includes "features that are produced from the interaction between constituent systems,"[33] it does not consider several artifacts in SoS modeling such as effectiveness and performance measures, risks, and dynamic environment of SoS.

In the military application domain, the mission is a strong concept that is defined in a rigorous manner and generally expressed through a well-structured document. The missions and means framework (MMF)[7] is a framework for defining the DoD military mission and evaluating its utility quantitatively. The MMF consists of 11 elements used to define military operations. Seven levels specify the mission: (a) mission purpose that defines the why of the military evolution and indicates the reason and purpose of the mission, (b) context and environment that define under what circumstances a mission is to be accomplished, (c) index and location/time that define the where in terms of geographic location and the when in terms of time, (d) tasks and operations that define the "do what" of the mission, and describes the implied tasks for mission accomplishment. The purpose of this level is to analyze the task outputs and subsequently evaluate the mission effectiveness, (e) functions and capabilities define the capabilities which enable forces to conduct operations, (f) components and forces that defines the "by whom" specification, represented with the military actors (integrated units, personnel, equipment, etc), and (g) interactions and effects that describe "how" the course of actions changes the state of components.

The military domain pays special attention to mission analysis in SoS SE. The mission context is unlike that in traditional SE approaches in which there is little flexibility because individual functions are mapped to only one element in the system.[5] According to Ref. 5, "mission context is a key element to assisting SoS engineers to determine the systems that must be involved and the functions that they must perform."

3.4 | Mission conceptual model

The analysis of the state of the art reveals that a novel approach, which incorporates the best practices of existing SE frameworks and approaches, taking into account the SoS artifacts and mission understanding, would be welcomed within the SoS SE community. The challenges addressed in this work are: the open SoS environment, the use of mission-oriented approach, and the definition of mission variants according the operational context.

To avoid any ambiguity, we introduce in the following section a mission conceptual model serving as the basis for our approach. The conceptual model was proposed based on the modeling experience using the SysML language of SE approaches, on SoS artifacts defined in Refs. 19, 26 and on the overall experience in rigorously defining a mission.[5–8] Figure 2 highlights the involved concepts.

In this paper, we define SoS as "a set of interacting systems that interact with each other and their environment to provide a common mission."[35] The *mission* is the main concept on the conceptual model (capability objectives artifact in Ref. 26). We define a mission as a finality that the SoS must achieve by collaborating constituent systems. We suggest decomposing the high-level mission (generally abstract) into more concrete missions. In the model, this aspect is expressed by the existence of the two classes *atomic* and *composite* and the relationship *refinedInto* between *composite* and *mission*. The refinement is stopped when we can identify the activity that is associated with the mission (CONOPS artifact in Ref. 26). An *activity* orders a set of *actions*; it can regroup *triggers* and *constraints*; and it can require *input parameters* and provide *output parameters* as in SE Approaches (SoS requirement space artifact in Ref. 26). A *role* handles *action* and gathers the required competencies (capability concept) to play the role needed to accomplish the action. The *capability* of a role is defined as "the ability to provide some expertise to the wider needs of an SoS"[30] (systems information artifact in Ref. 26). For each mission, the effectiveness measures must be determined (SoS performance data artifact in Ref. 26).

We define a role as "an abstraction of the characterization of the ideal behavior that will fulfill an action."[36] Several types of the constituent systems could be used to concretize a role in the concrete architecture: humans, hardware and software existing systems, and institutions. A constituent system is chosen when its capabilities match those required by a role and by considering the trade study and performance measures. The constituent systems can be integrated to meet a role capability. Measures must be defined for each role to guide the choices of constituent systems (SoS performance measures and method artifact in Ref. 26).

The nature of the collaboration between composite missions is basically described by the two variants of the mission composite: standard mission and mission with variation point. A standard mission is composed of sub-missions related with the AND, while a mission with a variation point is composed of mission variants (OR decomposition). The choice of a mission variant depends on the mission context (alternatives). The latter defines the circumstances under which a mission is to be accomplished.[7] We define the mission context as a set of contextual parameters that will determine the course of actions to be performed (performance data artifact in Ref. 26). These parameters are always updated; for example, a location data point is always updated by a geolocation system. The attributes of the mission metaclass allow the specification of the characteristics of the mission as a location if it is important, risk, etc. (SoS risks and mitigation artifact in Ref. 26).

The roles interoperate with each other (communicate, exchange data, etc.), and each role uses a set of interoperability mechanisms (Radio communication, ADSL connection, etc.). Interoperability mechanisms are present in an SoS; therefore, interoperability is possible by subtyping the communication media or by indirection via another constituent system. Interoperability thus does not require any ad hoc glue design. For instance, to manage crowd, the control and command center can obtain data from cameras via an Internet connection and can communicate with local authorities via a GSM network. Example is given in Section 6.

4 | GENERAL APPROACH

The proposed process is intended to support the analysis and architecture activities in the SoS wave life cycle. It offers a disciplined procedure for explicitly specifying the SoS end to end mission and generating the appropriate architecture. The process is applicable to acknowledged SoS in which the organization manages SoS and support the SoS SE while independent organizations and SE teams are responsible for the constituent systems.[12] Changes in the constituent

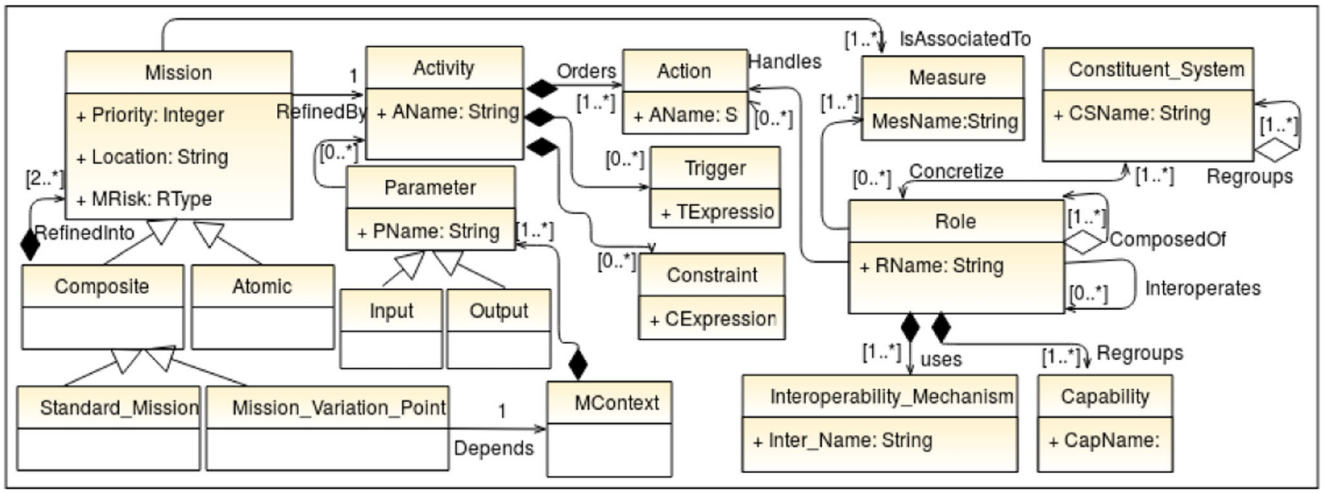


FIGURE 2 Mission conceptual model

systems are based on collaboration between the SoS and the system. The key ideas on which the process is based are as follows: (a) Relying on the design of and reuse of the SoS. (b) Bridging the SoS analysis and architecture development stages by taking advantage of the expertise of the ADE. (c) Automating the transition from the analysis stage to the architecture development stage as much as possible to avoid information loss between the ADE and the system architect. (d) Elaborate the concrete architecture model from the abstract one. The latter serves as an invariant that guides the choices of concrete entities.

In the SoS SE development environment, "two levels of stakeholders exist with mixed, possibly competing interests: the SoS stakeholders and constituent system stakeholders."[19] Since the constituent systems are independent and have their own objectives, stakeholders of individual systems may have little interest in the SoS, may assign SoS needs low priority, or may resist SoS demands pertaining to their system.[19] To manage the competing stakeholder interests, it is important for SoS SE engineer to focus on the operational view of the SoS and to balance the SoS objectives with the constituent system objectives.[19] We argue for the definition of two stakeholders to be involved in the development life cycle of an acknowledged SoS:

Application domain expert: The ADE masters the domain knowledge. Therefore, through her/his experience, this expert can anticipate the solution when refining the mission. The ADE focuses on the SoS operational environment (mission), and she/he does not control the constituent systems that impact the SoS but has the necessary knowledge to balance the SoS mission with constituent system goals. We propose the bridging of the analysis and architecture stages and automating the architecture synthesis as much as possible.

System architect: The system architect is responsible for the generation and realization of the architecture. Based on the conceptual models realized by the domain expert, she/he is responsible for deploying the required constituent systems to produce a concrete architecture.

Figure 3 introduces the main steps and the involved stakeholders in the process implementing the proposed approach. The latter is composed of top-down planning and decision making and bottom-up adjustment based on existing systems.

The goal is to refine the mission until the architecture is attained, while preserving the mission traceability. Therefore, the refinement steps are as follows:

1. Mission decomposition: This step is intended to provide a functional coarse grain view of the mission. This aspect is achieved through an analysis of the general mission objectives to recursively identify more precise sub-mission objectives. The criterion for stopping the mission decomposition is the identification of a process that may realize a given sub-mission. We developed a profile extending the SysML RE diagram to refine the main mission into sub-missions, create context-dependent variation points, and capture mission risks. Therefore, this step results in a mission functional model of the SoS.
2. Mission definition: The aim of this step is the design of a fine grained behavioral view of sub-missions using *activities*.The view is elaborated using the SysML activity diagram. Each sub-mission in the mission functional model is associated with an activity using the refine relationship. Complex activities can be decomposed into sub-activities, and the criterion for stopping activity decomposition is when a subactivity corresponds to a role capability that we call an action.
3. Role definition: The *role* is used to provide an abstract representation of hierarchy of entities having capabilities that enable the achievement of the mission. The capabilities could be provided or required by roles, thereby allowing the composition of roles. The produced model for role definition is based on a SysML profile extending the BDD.
4. Role assignment: As mentioned previously, activities are composed of actions that correspond to role capabilities. The role is composed of several capabilities, and the same capability can appear in different roles. Therefore, this step is intended to designate the role that must be associated with each action of an activity. This

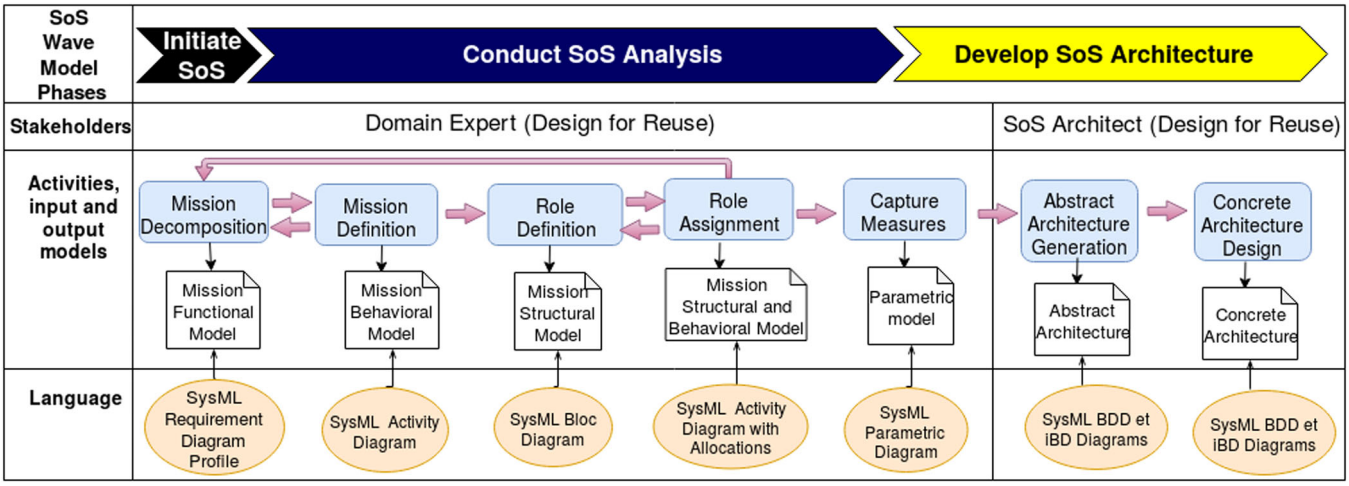


FIGURE 3 Actors and responsibilities

association creates a link with the constituent system through the assigned role.

5. Capture measures: In this step, several effectiveness measures are required to be expressed using the SysML parametric diagram. On one hand, the mission effectiveness metrics are defined by the ADE for assessing the overall performance of the SoS mission. On the other hand, the metrics for choosing the best system among the existing ones for playing a role are defined for each role.
6. Abstract architecture generation and concrete architecture design: The architecture is a structural view that describes the constituent systems of the SoS and their connections. However, all the above-mentioned definitions refer only to roles instead of constituent systems. Therefore, the first generated architecture from the given definitions corresponds to the abstract architecture of the SoS. Thus, the abstract architecture is progressively refined during the architecture analysis to get the concrete architecture. For this step, both the SysML internal block diagram and SysML BDD are employed in this phase.

5 | MISSION AND ROLE MODELING

In the proposed approach, the ADE is responsible for the structural and behavioral model. Constructing such a model is realized through an iterative process. The process is stopped when the expert finds a compromise between the capabilities needed by the mission and those offered by realistic constituent systems. The mission and role modeling is detailed in the following subsections.

5.1 | Mission decomposition

This phase is intended to understand SoS top level missions and to plan a mission strategy. The essential elements considered in this phase are description of the main missions, variation points, mission location, mission risk, and priority. We propose the gradual functional decomposition of the SoS mission and the splitting of complex missions into simple ones.[37,38] This task is made possible by using the SysML RE. The RE diagram "allows the specification of a function that a system must perform or a performance condition that a system must achieve."[39] The SysML RE provides modeling constructs to represent text-based requirements and relate these requirements to other modeling elements. Different relationships are furnished to allow relating requirements to other requirements or to other model elements. These relationships include relationships for "defining a requirements hierarchy, deriving requirements, satisfying requirements, verifying requirements, and refining requirements."[39] A standard requirement includes the unique identifier and text requirement. Users can add properties if needed.[39]

The basic SysML RE is not sufficient to describe all the concepts cited above. For instance, it cannot represent the mission priority and mission risk. Furthermore, it does not allow the creation of variation points since the semantic of decomposition is the conjunction. Therefore, we propose the extension of the RE diagram to allow the ADE to add the desired properties and variation points. This extension is possible since SysML is a highly extensible modeling language.[39] A stereotype is one of the types of extensibility mechanisms in SysML; it is a profile class that allows designers to extend the vocabulary of SysML to create new model elements, which are derived from existing ones but have domain-specific properties.[39]

Figure 4 illustrates the extension of the SysML RE. The default properties *id* and *text* specify the unique identifier and text requirement, respectively. The *Requirement* is a stereotype that inherits from the metaclass *Class* of UML. The extension is performed by creating a stereotype called *Mission*, which contains the added properties. The stereotype Mission inherits the properties of its superstereotype Requirement, and the following properties are added: *location*, *risk*, *priority*, *version*, and *date*.

The default property *text* of the stereotype *Requirement* can be used to describe the mission goal. The mission location may represent an IP address, GPS coordinates, polar coordinates, region, etc. For this reason, the location is considered as a string parameter. The successive refinements of the main mission generate several sub-missions. The

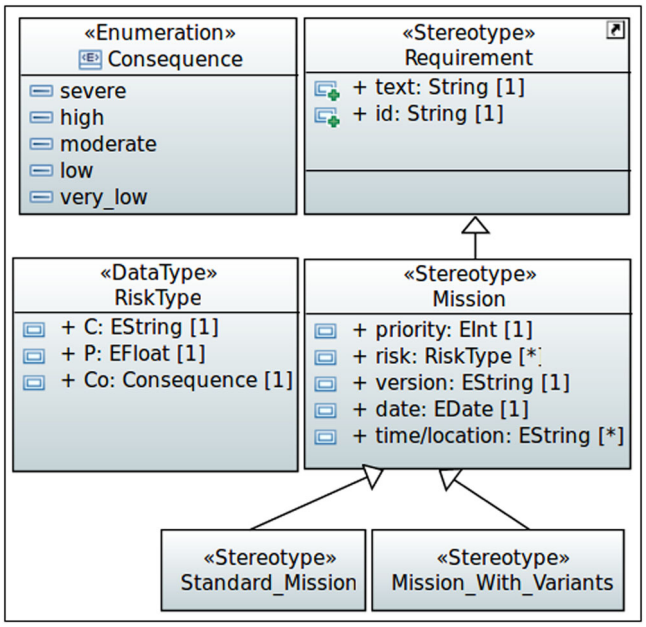


FIGURE 4 Mission stereotype

status of these sub-missions is not the same under the main mission, and the priority is the parameter that indicates the importance of each sub-mission. We assume that the priority can take an integer value that indicates the relevance degree of the mission. Given the context uncertainty, risks can affect missions. A risk must first be identified and later mitigated if possible. Alternative activities are defined to prevent the consequences from occurring. Based on the definitions of risk[40,41], we propose the consideration of risk by attaching the triple R = C,P,Co to each mission (see Figure 4) in which C is a string value representing the future risk cause, and P is a numeric value representing the probability of risk occurrence. The suggested values for Co that represent the consequence are severe, high, moderate, low, or very low.

The wave model is based on SoS upgrade cycles. Therefore, the properties described above can change in each upgrade cycle. For example, the priorities are reassessed in each cycle. Therefore, we use the mission stability level as a parameter to determine the stability of the mission definition. We referred to Ref. 42 to define this parameter, in which the authors propose to use the version and date of creation/change of/in properties to indicate if and when the mission was changed. To make our process applicable to a wide range of practical contexts, we propose supporting mission variants in the decomposition of the mission. To this end, we define two new stereotypes called the *Standard Mission* and *Mission with a Variation Point*, which inherit from the stereotype *Mission*. The *Standard Mission* is composed of a conjunction of sub-missions while the *Mission with a Variation Point* is composed of a disjunction of sub-missions. When defining missions, the *Mission with a Variation Point* must be defined with the contextual information that allows the resolution of alternatives (see Figure 5).

To match the activities/actions to each mission, we propose using the *refine* relationship in the RE diagram. The refine relationship is used to relate a mission to another model element. We use the *call behavior action* element that references an activity to refine a mission. A mission is refined by an activity, and the referenced activity is described later using an activity diagram (see Figure 5).

5.2 | Mission definition

This phase is intended to define the activities, course of actions, and capabilities required to handle actions considering the variability in the user environment that impacts the ways the capabilities are executed. Once the activities have been identified, they are described in this phase using the SysML activity diagram. Each *call behavior action* element identified in the mission functional model is refined using an activity diagram. The activity refinement process is stopped when the expert reaches a process composed only of actions that correspond to a capability. In the SysML activity diagram, partitions are used to group actions that have some common characteristics.[17,39] Partitions are commonly used to regroup actions that are performed by the same system. We propose using partitions to group all the actions that are performed by the same role.

The mission strategy may change according to the context. We described the SoS context using activity *entry parameters*. In this manner, the context of a mission can be inferred from the parameter values, and all the alternatives can be defined at an early stage. *Signals* are used to express the mission triggers, while action scheduling can be expressed using *activities*, *actions*, *data*, and *control flows*. *Constraints* can be set on actions to specify the business semantics. The mission definition phase results in the fine grained behavioral view of the sub-missions, called the mission definition model, which is defined using the SysML activity diagram.

5.3 | Role definition and assignment

The role definition phase focuses on constituent system level information that impacts the SoS mission. The roles, capabilities, and the possible constituent systems are explored. Given the uncertainty of SoS boundaries, the challenge is to include the roles that can support the SoS mission, and to exclude the useless role ones. In our approach, the ADE considers roles that she/he judges to be relevant entities to the SoS mission. A constituent system enters the SoS boundary when it begins to affect the SoS behavior and leaves when its contribution is negated.[13]

Role modeling can be performed using a BDD, in which the structural and behavioral features of a role are described. Hierarchical relationships between the roles can be defined. The capabilities are modeled using operations. Coherent set of capabilities are grouped into interfaces. A *realization dependency* is added from the role to each provided interface, which means that the role will provide each capability in that interface. A role can assert that it requires a set of capabilities by adding a ≪uses≫ dependency to an interface. We created a stereotype Role that inherits the properties of its superstereotype *block* to use SoS vocabulary domain. The communication between roles is done by subtyping their communication media. For example, information exchange could be done by the use of Radio communication.

The purpose of the role assignment phase is to allow the expert to determine the constituent systems that will fulfill actions. Therefore,

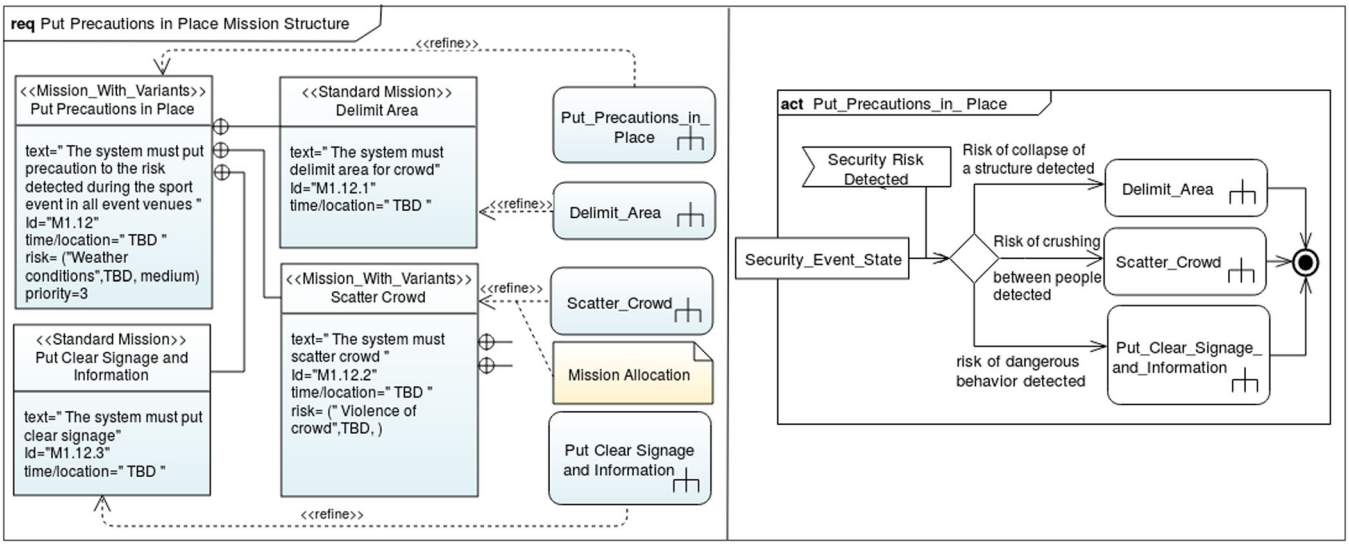


FIGURE 5 Mission decomposition and allocation example

she/he provides a model that combines the functional and structural views of the mission. As shown in Figure 6, role assignment is based on the functional allocation of activity partition into a role. The role must have capabilities that allow it to achieve all actions contained in the partition. Actions are allocated to capabilities using a call operation action, as shown in Figure 6.

5.4 | Capture measures

This phase focuses on the performance of the SoS solution. The aim is to enhance the SoS performance as much as possible. As in OOSEM, we propose to capture MoEs in the SysML BDD, and the methods and models for calculating the MoEs are described using the SysML parametric diagrams with the objective function. Two kinds of measures must be adopted in this phase:

- Mission MoEs: The mission MoEs represent the mission-level performance parameter whose value is critical for achieving the desired mission effectiveness. Ref. 17 proposed a useful technique for deducting the MoEs, which consists of using a fishbone diagram to represent a tree of cause-effect dependencies, and then build the parametric diagram from the cause-effect tree.
- Roles measures of performance: Role MoPs are parameters captured to choose between several candidate constituent systems that can concertize a role. Several criteria can be considered, namely, the availability, cost, performance, etc.

5.5 | Abstract and concrete architectures

The next step in our approach is to generate the SoS abstract architecture. The focus of this phase is describing which abstract roles interact within a configuration and how. We propose automatizing the process of generating the abstract architecture from the activity diagram and the roles BDD. This abstract architecture is defined using the SysML BDD and IBD. The BDD defines logical decomposition of the main block into roles, and the IBD defines the interactions among the roles such that they satisfy the mission. In the BDD, a block is created for each decomposition of the partition hierarchy. The block features are captured from the role BDD (see Figure 6).

Figure 6 shows also traceability among the different models. The *refine* relationship is used to trace the missions to the corresponding activities. Activities orders actions that are represented by *call behavior actions*. The later correspond to operations offered by roles. The IBD captures the internal structure of a block in terms of the parts, properties, and connectors, and this structure is used to display different connections between the parts (roles) that compose the block. The main idea of the transformation here is to consider the activity diagram as a starting point. Considering that an activity refines a mission, our main goal is to build the block that can satisfy this mission (see Figure 6). To this end, we associate each activity model element to a block element that has the name of the model; this block is considered as the main block.

Since our activity diagram is composed mainly of partitions allocated to roles, the partitions are represented as parts in the main block. The parts have the same names and types as the partitions. An activity is characterized by input (or output) parameters. The parameters are provided (or required) by other blocks, which mean that each parameter corresponds to an interaction point that is represented in the IBD by a block port. Each flow between the actions from different partitions indicates that data exchange occurs between two different roles. This aspect means that a flow between the two corresponding parts must be created using the corresponding ports. The elements of the resulting IBD are allocated since they are generated from the activity diagram. The resulting IBD is consistent with the activity diagram, as shown in Figure 6.

The abstract architecture is used to obtain one of the possible concrete architectures. To define the concrete architecture that will satisfy the mission, we used the synthesis candidate physical

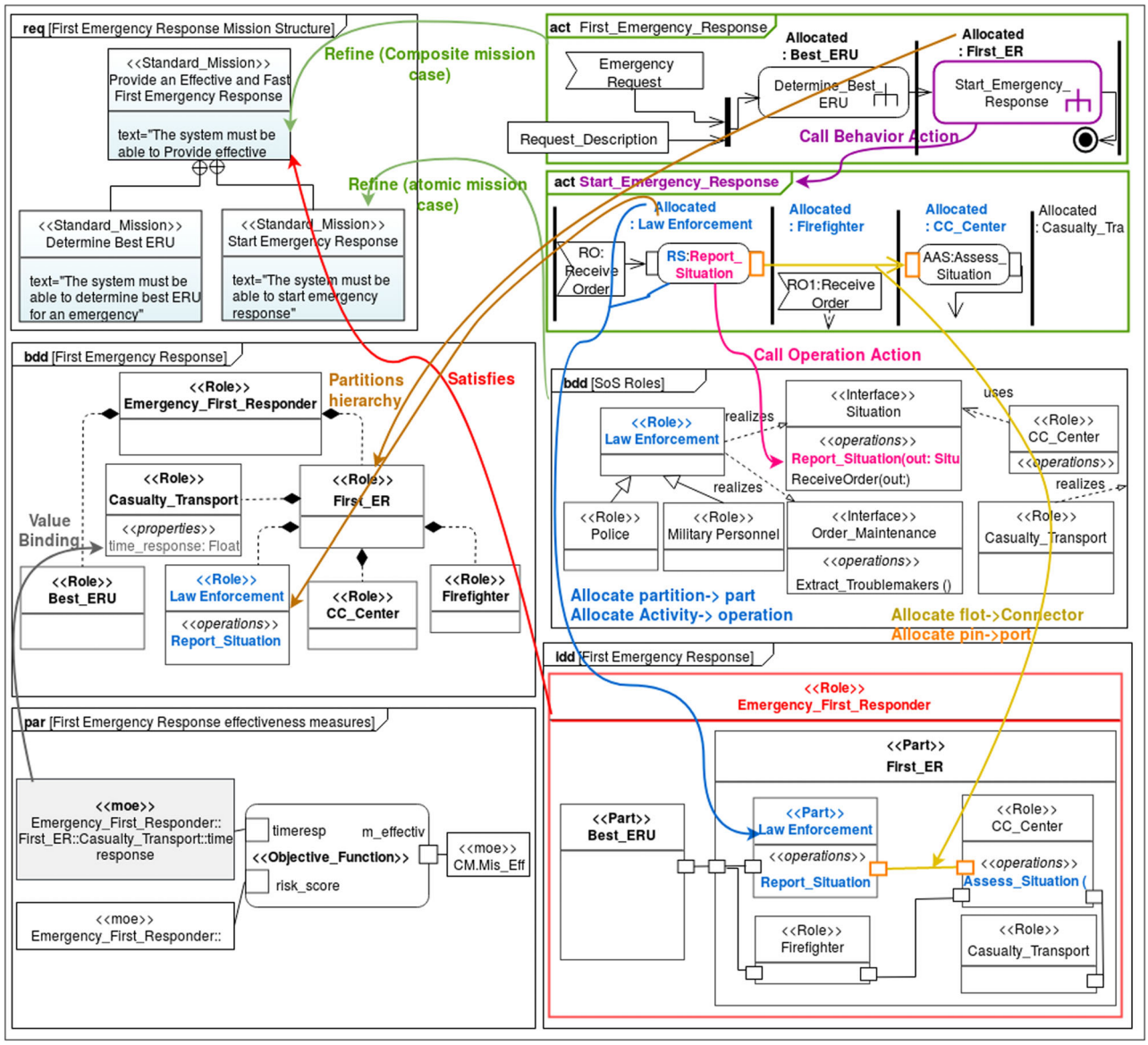


FIGURE 6 Role assignment principles

architectures OOSEM's activity[17] since it supports a geographical distribution of components, which is an important aspect of SoS. This OOSEM's activity defines the concrete architecture in terms of its physical components and relationships, and their distribution across system nodes. The physical components of the system are represented by hardware, software, and persistent data. We propose that humans can also be used as physical components. The system nodes represent a partitioning of components based on partitioning strategies (physical location, etc.). A concrete architecture is defined by associating each role to combination of human, hardware, and software constituent systems. The MoEs are used to select the preferred architecture.[17]

6 | CASE STUDY

In this section, we discuss and explain the steps of the proposed process using a case study focusing on an SoS dedicated to crowd management during a football event. This case study is an SoS which is defined in Ref. 43. It is in the same time part of the disaster response system of systems, which is a widely used example of SoS.[4,32] This SoS is aimed at developing an integrated crowd control system during temporary events of mass transit, such as sports events or political meetings. The case study objective is to refine the crowd management SoS capability objectives into an architecture description using the mission paradigm. The case study data are collected primarily using document analysis based on documents concerning a French governmental field for crowd management and emergency response[44] and the FIFA regulations.[45]

6.1 | Mission decomposition

The initial top-level mission for the Crowd Management SoS is to *maximize the safety of crowd, property and event venues, by minimizing risks and providing emergency response while managing costs during all sport*

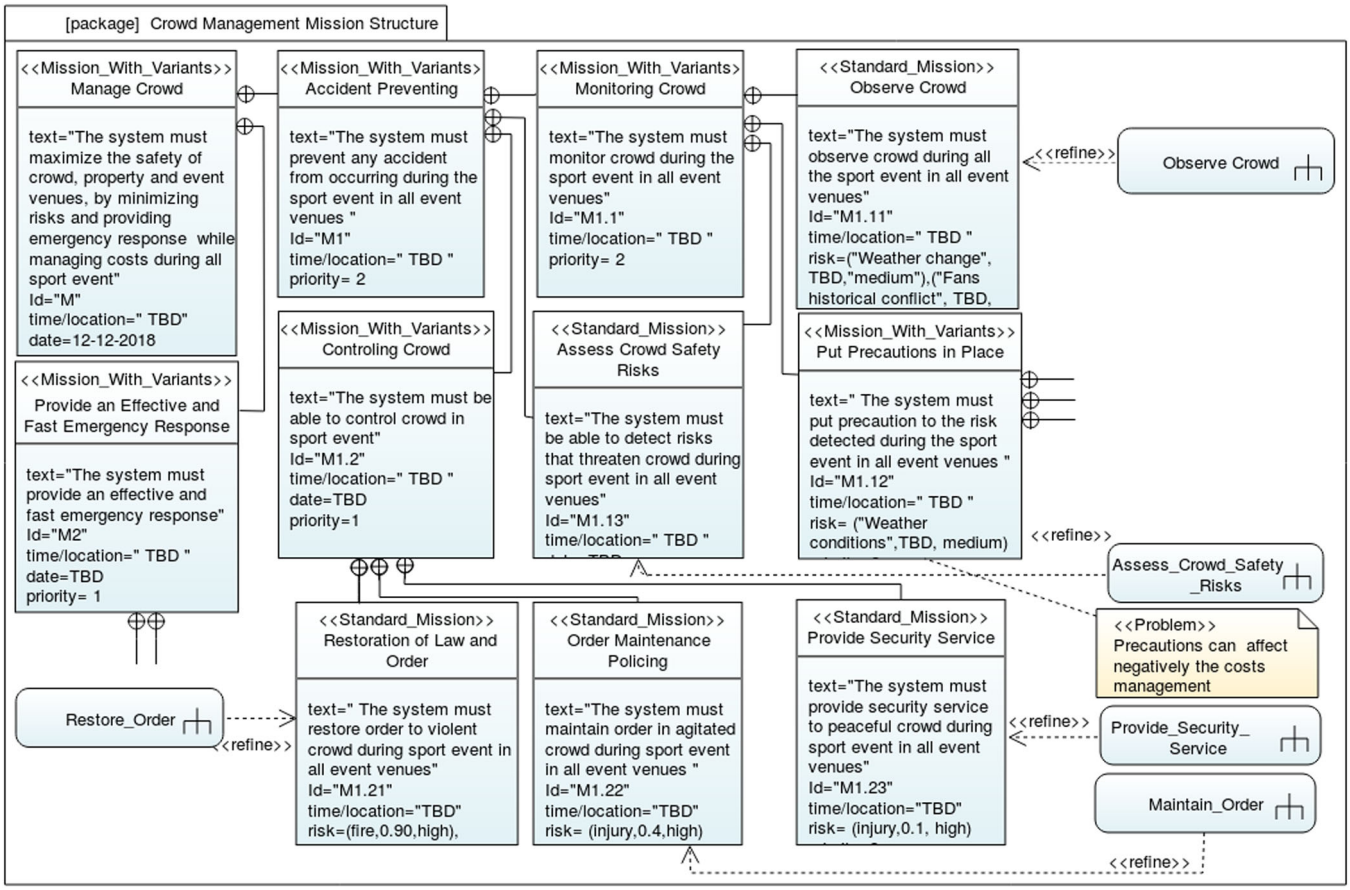


FIGURE 7 Mission decomposition diagram

event. It has to be decomposed into several sub-mission. Decomposition is contained in the Mission Structure package. Figure 7 shows part of the realized Mission Functional Model. Initial mission is represented by the *Manage Crowd* mission. It includes the text statement describing the mission and the "*M*" id. Two second-level sub-missions are designed : *Accident Preventing* and *Manage Costs* ones where the first one has higher priority than the second one. In order to achieve the *Management Crowd* mission, the two sub-missions must be realized. It is specified by the *AND* operator. The mission time/location are not determined yet and depends on the event location. Thus we used the *TBD* acronym to indicate that the values will be determined (To Be Determined). The decomposition process is iterative. For instance, the *Accident Preventing* sub-mission is decomposed into *Controlling Crowd* and *Monitoring Crowd* sub-mission. The later is next decomposed in several sub-missions. Decomposition is stopped when we can refine a sub-mission by an activity using a refine relationship. Using SysML *refine* relationship allows to reuse the SysML traceability mechanisms. The *Observe Crowd in Normal Conditions* Call Behavior action is used to refine the *Observe Crowd* sub-mission, by adding a set of activities description. Each one encompasses a set of activities/actions that refine the sub-mission and take into account the corresponding risks.

The initial top level mission for the crowd management SoS is to *maximize the safety of the crowd, property and event venues, by minimizing risks and providing emergency response while managing costs during all sports events*. This mission must be decomposed into several sub-missions. The decomposition is a part of the mission structure package. Figure 7 shows part of the realized mission functional model. The initial mission is represented by the *manage crowd* mission, which includes the text statement describing the mission and the "M" id. Two second level sub-missions are designed, namely, *accident prevention* and *provide an effective and fast emergency response*, and *accident prevention* has a lower priority than the second one.

The *crowd management* mission is a mission with variants. The semantic of decomposition is the OR operator since the *provide an effective and fast emergency response* mission is not executed if there is no emergency request. The mission time/location are not determined yet and depend on the event location. Thus, we use the TBD acronym to indicate that the values will be determined later. The decomposition process is iterative. For instance, the accident prevention sub-mission is decomposed into crowd control and crowd monitoring sub-missions, and the crowd monitoring sub-mission is further decomposed into several sub-missions. The decomposition is stopped when we can refine a sub-mission by an activity using the refine relationship. Using the SysML refine relationship allows the reuse of the SysML traceability mechanisms. The observe crowd call behavior action is used to refine the observe crowd sub-mission by adding a set of activity description. Each activity encompasses a set of activities/actions that refine the sub-mission and take into account the corresponding risks.

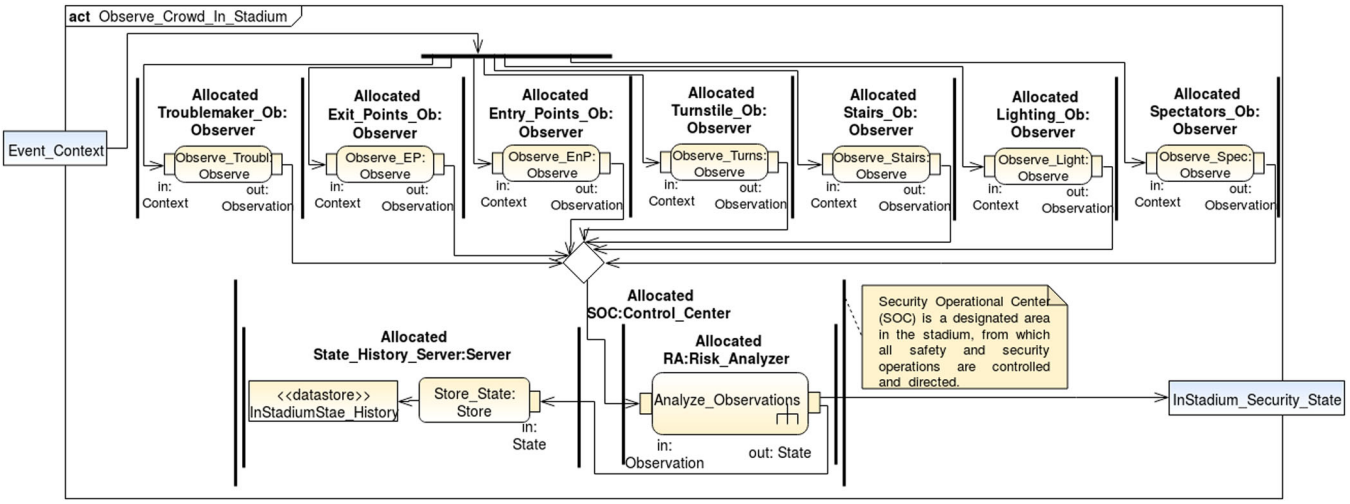


FIGURE 8 Mission definition diagram

6.2 | Mission definition

For each *call behavior action* refining a sub-mission in the Mission Decomposition Diagram, an activity is created with the same name. This aspect ensures that each mission is associated with a set of ordered actions. The activity *Observe Crowd* refines the *Observe Crowd* mission. The activity that realizes the *Observe Crowd* actions when a high risk of conflict is identified is based on observation actions inside and outside the stadium. Figure 8 shows the activity that realizes the *Observe Crowd in Stadium* actions. Each action is stored in a partition block, and the partitions are allocated to the corresponding roles. *Observe Entry Points*, *Observe Troublemakers*, or *Observe Spectators* are examples of observation actions that provide observations as outputs. The *analyze observation* activity retrieves the observations and generates a state of safety and security in the stadium. The *analyze observation* activity is represented by *call behavior action* (see Figure 8), which means that it is associated with an activity diagram that describes it.

The observation actions should take into account several factors such as political tensions, historical rivalry between fans, and supporter profiles. Each contextual information is given as an input by the activity parameter (see Figure 8). The expected result from the observe crowd with historical conflict between fans activity is the secure event state that is provided as an output parameter.

Mission Decomposition Diagram is not supposed to contain partitions allocations. To avoid presenting two similar diagrams, one with the allocations and the second with no allocations, we have presented only Mission Decomposition Diagram with allocations.

6.3 | Role definition and assignment

Once actions/activities have been defined, they must be attributed to a role. A role can be abstract to varying degrees and can be specialized using the inheritance relationship. Figure 9 shows part of the crowd management Role BDD, which includes a set of role hierarchies from the most abstract role to concrete roles. For instance, according to the situation, cost, service availability, and observation target, the *observation* action could be performed using a *camera*, *steward*, *smoke detector*, etc.

As shown in Figure 9, the role observer provides a capability *Crowd Observation*. *Crowd Observation* is required by the risk analyzer. The assignment of roles to actions is performed by typing partitions with roles. For instance, in Figure 8, the safety equipment observer partition is typed by the observer role, because it is not possible to determine at this stage if a camera will exist in this place or another physical entity will be used.

The observer role provides a capability *Crowd Observation*, which includes an operation called *observe*, as shown in Figure 9. A call operation action for *observe* is shown with pins corresponding to the entry and output parameters with all observation actions; for instance, *Observe Troublemakers* and *Observe Spectators* call the operation observe, as shown in Figure 8. Allocating roles to partitions allows the maintenance of traceability between the role base and the activities/actions related to a sub-mission.

Figure 10 shows the mission-level performance and effectiveness measures that are based on mission outcomes. A part of the MoEs of the *crowd management in sport event mission* are: the supporters satisfaction, the risks detected and avoided, operational availability, and the global cost. The value of each MoE is also calculated. For example, the risk score is calculated using objective function from the max density, cross flows emplacement and number, weather conditions, etc. The mission-level effectiveness is performed to support the evaluation of the design solution.

6.4 | Architecture design

Figures 11 and 12 show, respectively, an aggregate of roles, where each role achieves a specific *Crowd Observer in Stadium* activity or action, and the interconnection among parts that participated in the *Crowd Observer in Stadium* activity. The two diagrams are generated automatically using the ATL rules from the Crowd Observer activity diagram and the role diagram.

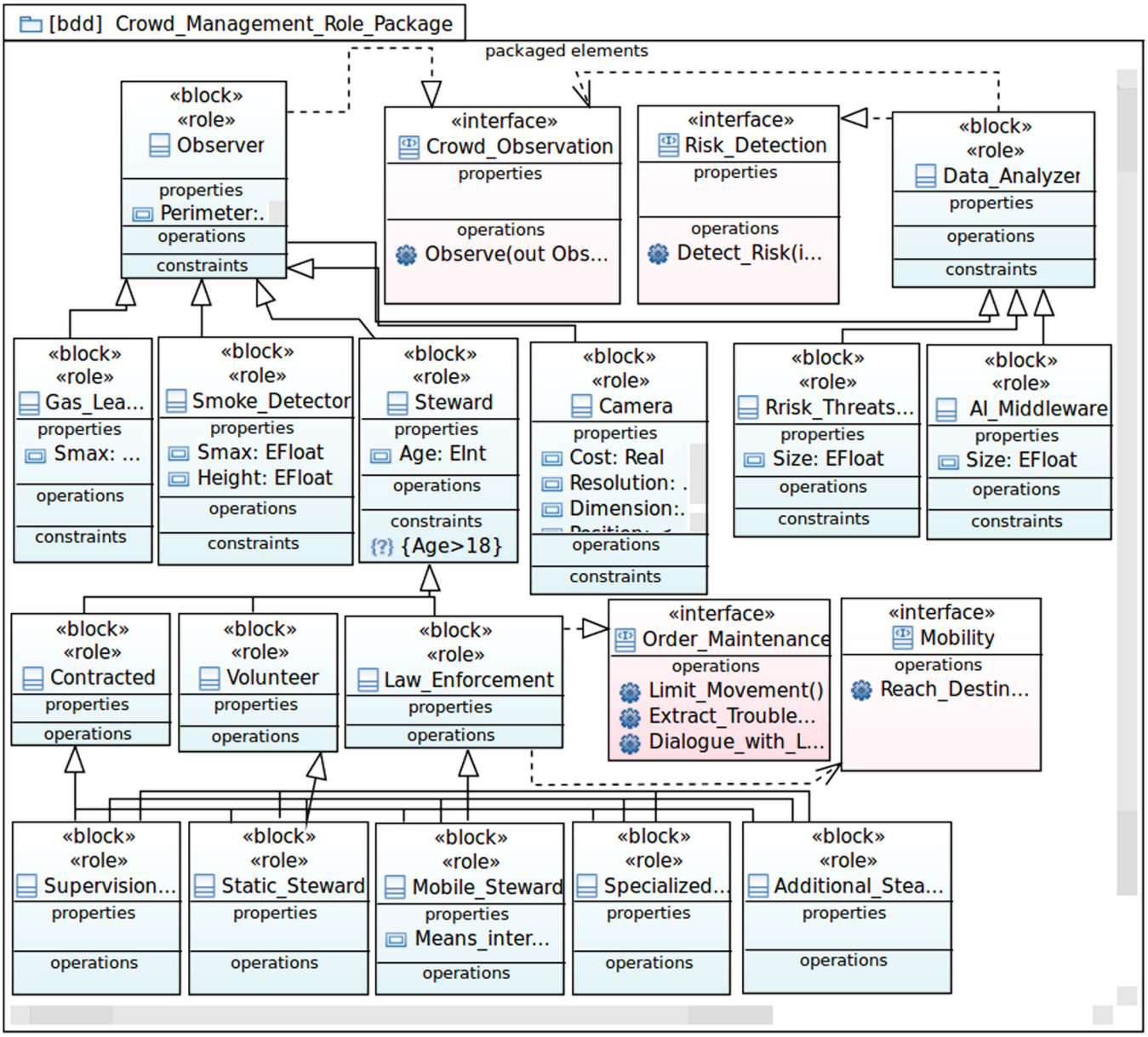


FIGURE 9 BDD roles diagram

In the BDD (see Figure 11), the *In Stadium Crowd Observer* role aggregates the *Control Center* role and the seven *Observer* roles (troublemakers observer, entry points observer, etc). The seven observer roles are responsible of the *observe* mission at a particular location.

In the IBD (see Figure 12), the parts represent how the roles are used in the observation context and have the same role names as shown in the activity diagram. The flow ports are consistent with their definition in the activity diagram. The IBD for *In Stadium Crowd Observer* role shows the interconnection among the roles that are involved in the *In Stadium Crowd Observer* Activity Diagram. However, there is additional activity diagram that corresponds to the *Analyze Observations* activity. This activity diagram includes different sets of interacting roles. Indeed, all the parts (roles) from all the activity diagrams are represented in Figure 12. Likewise, the hierarchy of all roles is represented in Figure 11. The *Control Center* aggregates the *Server* and *Risk Analyzer* roles. The *Risk Analyzer* role aggregates the *Receptor*, *Recorder*, *Data Analyzer*, and the *CCTV Operator*.

The abstract architecture is used to obtain a possible concrete architecture by replacing the abstract items with concrete ones. Each solution is characterized by a set of attributes that have a value distribution. The attributes for a given solution are then evaluated using an objective function, and the results for each alternative are compared to select the preferred solution. Figure 13 shows two variants of the *Observer* role serving as solution to perform the observer role in the *observe exit points* mission. The operational availability, cost, and security effectiveness are the MoE for the observe mission. The overall effectiveness is calculated for each alternative using a weighted equation of their MoE values.

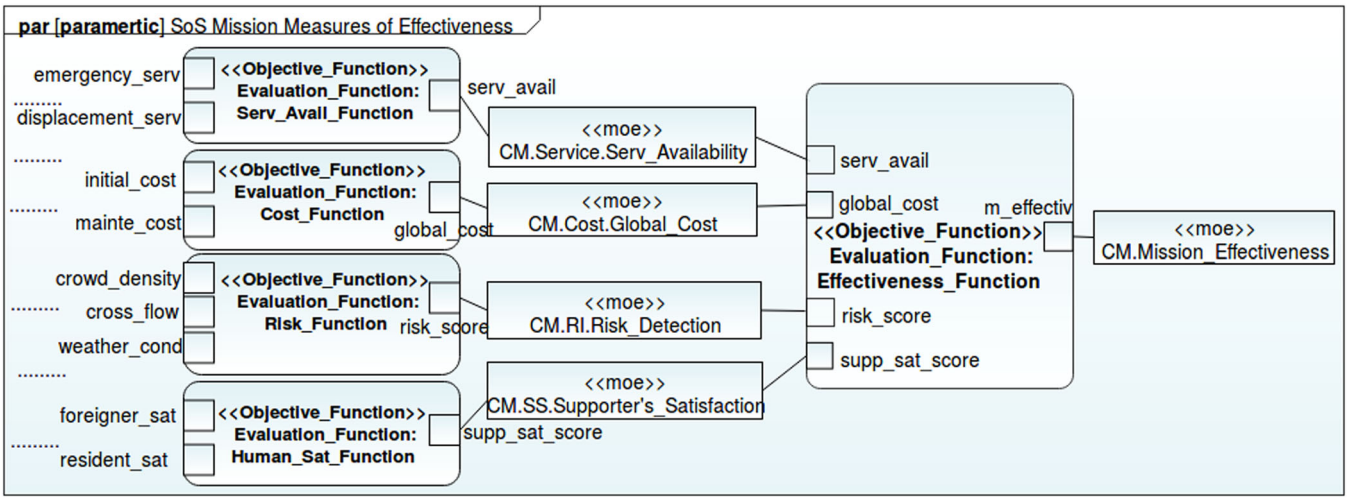


FIGURE 10 Mission effectiveness measures

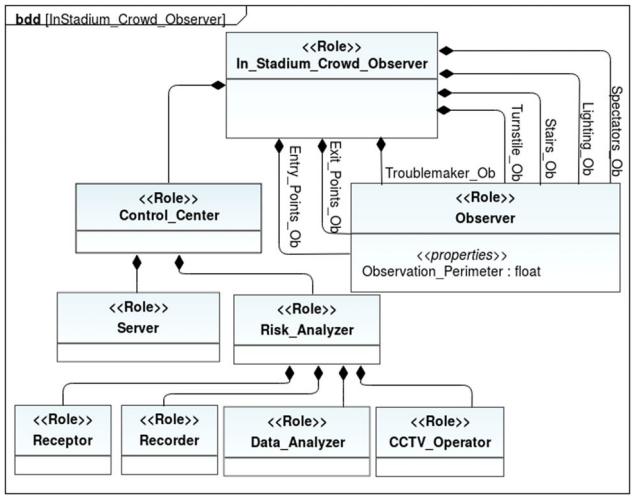


FIGURE 11 Crowd Observer in Stadium abstract BDD

According to the effectiveness of each alternative. The architect selects, evaluates and chooses the preferred concrete constituent systems. Like OOSEM, we used the concept of "physical node that represents an aggregation of physical components at a particular location." Figure 14 represents an example of concrete roles allocated to the abstract roles. The *Troublemakers Observer* role was assigned to *mobile steward* human concrete role, who are engaged to penetrate the crowd and observe troublemakers. The concrete architecture constrains the solution space with preselected concrete systems that are available and are able to be assembled in the SoS. When a role is allocated to software, the later must also be allocated to a corresponding hardware role to execute it. Likewise, when a role is allocated to human, the later must also be allocated to a corresponding hardware role to allow him communicate. As already mentioned and as shown in Figure 14, the steward can communicate using using the Radio headsets.

The concrete architecture evaluation is performed using the overall mission effectiveness (see Figure 10). Simulation remains the best way to analyze the impact of an architecture solution on mission mission effectiveness.

6.5 | Discussion

The main goal of this paper is to consider mission thread to bridge the dissociation between SoS objectives, and the individual functionalities undertaken by constituent systems, to support the SoS mission. To address this SoS SE challenge, we proposed in this paper to maintain a mission focus throughout the SoS SE analysis and the architecting process included in the wave model.

6.5.1 | Back to SoS SE challenges

This section returns to a subset of challenge problems that were introduced in Section 2, and considers how each challenge is addressed based on the models that have been developed in the case study.

- Operational independence: Which means that any constituent system is independent and can operate serviceably if the SoS is disassembled. In the proposed process, we considered that a constituent system is an independent entity that is able to provide to the SoS a subset of its functionalities, which are called capabilities. The mission actions are allocated to roles and the roles are replaced by available constituent systems based on performance measures. Thus, when a constituent system is disassembled from the SoS, it continues to operate independently.
- Dynamic environment: Constituent systems supporting each role in a mission will vary over the course of the actions. The mission context is key player to determine the constituent systems that are involved in mission accomplishment. We use set of parameters to determine the mission context, when parameters change, the course of actions in the activity diagram changes. The activity diagram in Figure 5 shows a decision node that depends on the security event state. The activity that will be executed depends on the value of risk. Indeed, different architectures are generated for different risk values.
- SoS evolution: SoS evolves over time, but evolves slowly. The evolution could consist of the addition of a new constituent system or a

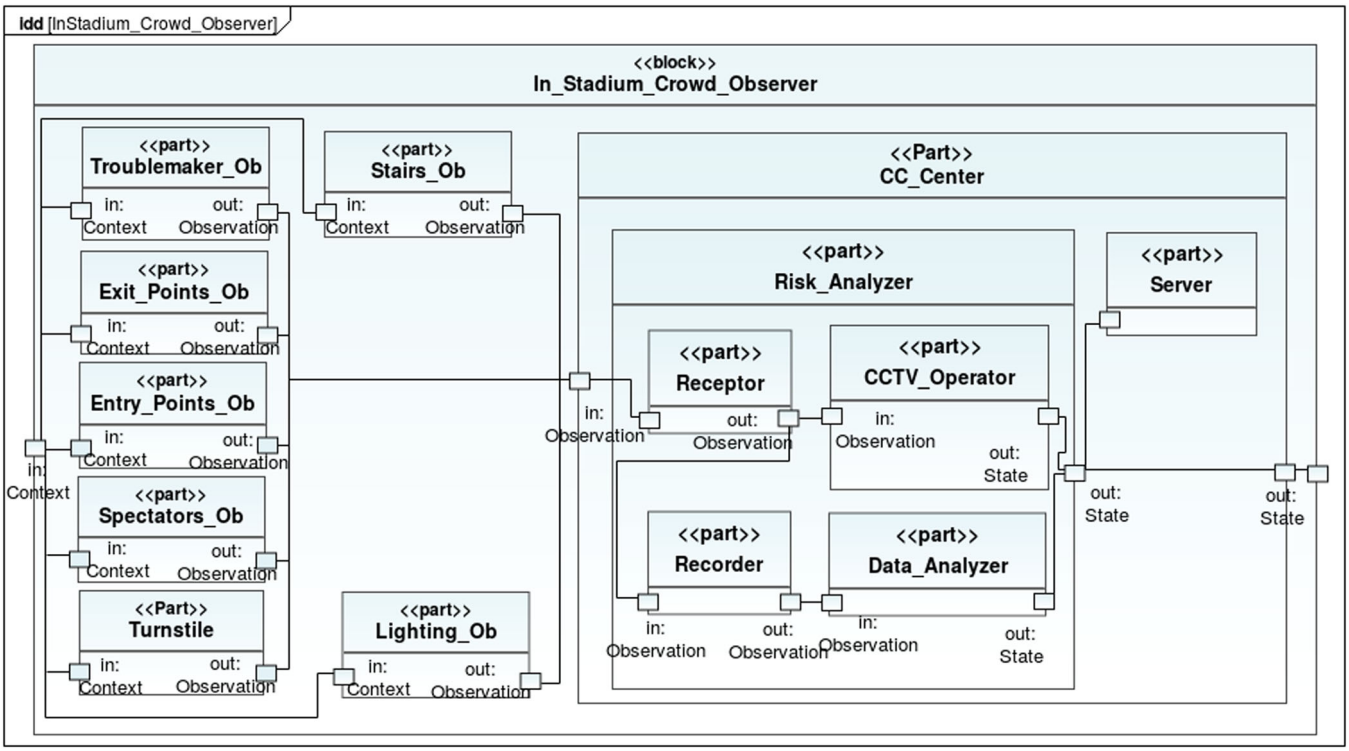


FIGURE 12 Crowd Observer in Stadium abstract IBD

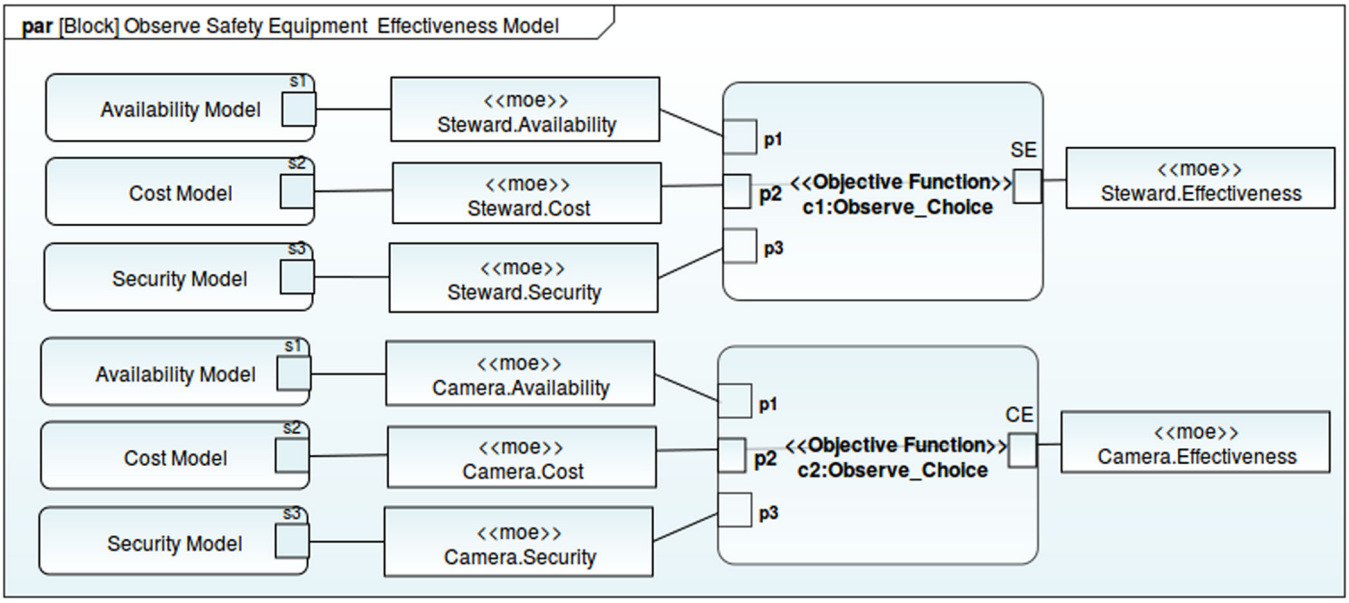


FIGURE 13 Effectiveness evaluation results between the two observers variants

change in the behavior of a constituent system. In the case of addition, we add the new constituent system in the role diagram and add the generalization relationships toward the appropriate roles, or create a new role if the new constituent system holds a new behavior. In the case of a change in the behavior of a constituent system, the role diagram is also updated according to the new capabilities. In the two cases (addition or modification of constituent system), the measures of performance are made and compared to the MoP of the constituent systems holding the same capabilities. The challenge here for the architect is to propagate changes across the concrete architecture.

6.5.2 | Back to the state of the art

SE approaches are mature and guide engineers in the analysis, development, and documentation of complex systems. The majority of them

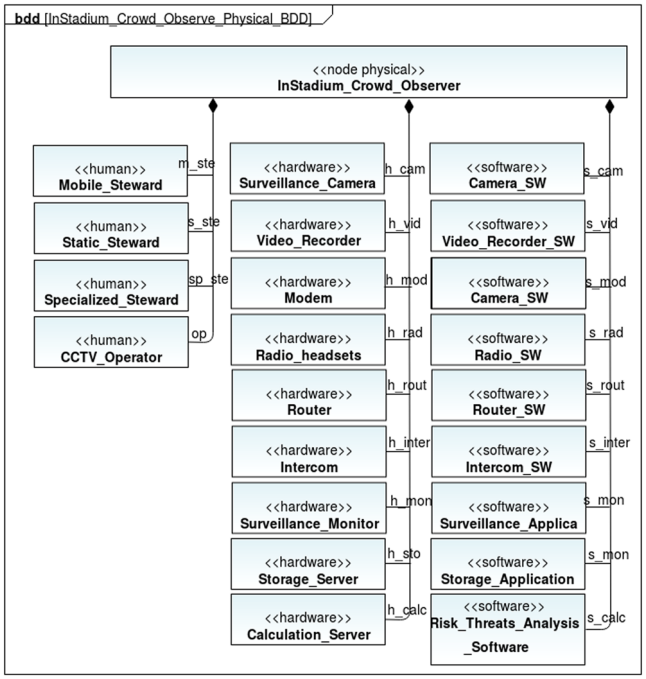


FIGURE 14 Physical block definition diagram of Crowd Observer in Stadium node

are based on SysML to take advantage of its syntactic richness. We took advantage of their maturity from the methodological point of view: We captured performance and effectiveness measures using the parametric diagram as in OOSEM. For the role definition phase, we used the service-based interactions provided and required as Harmony SE does to describe the system components. We have refined the RE diagram directly by an activity diagram as in MagicGrid. However, while these approaches are based on the concept of *requirement* that must be met, our approach is based on the *mission* paradigm. Through this paradigm, we want to offer a more operational view that supports all the tactical information that can change the order of execution of the actions, the involved systems, and different operational environments. Thus, we considered in the decomposition through the RE diagram the different points of variation of a mission, something that is not considered in the SE approaches. These points of variation are solved by activity diagrams taking contextual information as input parameters to show the different alternatives and decisions. So refinement does not just consider leaf missions as in other approaches.

SoS SE work is most often influenced by DoDAF and MoDAF frameworks. For example, the Compass and DANSE projects have tried to reduce the number of views, proposed by these frameworks, to make them more manageable. An example of these influences concerns the fact of considering concrete systems during the specification stage. So, SoS boundaries are dictated by existing systems. In our approach, the SoS specification is more open thanks to the use of a more abstract definition of constituents (Role concept).

Due to the diversity that can exist between the constituent systems, interoperability becomes an important aspect when describing an SoS. We consider that this aspect is the responsibility of the system architect and must be treated in a downstream step with respect to the specification. This is out of scope of this paper and will be dealt with in a specific work.

7 | CONCLUSION

This paper supports the maintenance of a mission focus throughout the SoS SE analysis and the architecture process included in the wave model. In fact, the SoS are acquired to satisfy new capabilities in a mission context. The later is a key element to assist SoS engineers to determine the systems that must be involved and the functions they must perform.

The first contribution of this paper is the proposition of a mission conceptual model. The later shows the main concepts characterizing SoS mission. The mission conceptual model was proposed based on the modeling experience using the SysML language of SE approaches, on SoS artifacts defined in Refs. 19, 26 and on the overall experience in rigorously defining a mission.[5–8] The second and the main contribution of this paper is the proposition of a mission-based process that strengthens the links between the SoS analysis stage and the architecture stage in the SoS wave life cycle. The process concerns an acknowledged SoS.

The SoS analysis stage is conducted by the ADE, and the architecture stage is directed by the system architect. The ADE defines the SoS mission taking into consideration the mission context. The mission context is represented by global parameters, their corresponding values determine the mission threads. The ADE defines also the roles, abstract entities that encapsulate the ideal behavior that will fulfill an action. The concept of role is used to deal with the uncertainty of the availability of SoS constituent systems. We used model transformation mechanisms to generate the corresponding abstract architecture, from which the system architect can deploy concrete constituent systems. Different abstract architectures are generated for different mission contexts, and different possible concrete architecture could be obtained by replacing the roles with human, software, and hardware constituent systems. Measures of performance where used to choose the best constituent system and the available one.

The use of this approach to model several SoS in the same domain can help identify recurrent concepts in the form of parts of submissions, roles, and capabilities. Recording these concepts in knowledge bases allows their reuse, which can help the domain expert enhance the efficiency in SoS design and decrease the cost of using the model-based approach. We consider the SysML models as the basis for the assessment of the SoS architecture. Such models allow to assess gaps in mission performance and to improve the analysis of SoS behavior earlier in the development cycle.

Dahmann et al[46] argue that the SysML model "represents an unambiguous, structured and executable representation of the SoS architecture that can be exploited and simulated." The simulation is useful to enhance the effectiveness of SoS mission or to observe the SoS behavior in order to detect undesired emergent behavior.[47] This implies to make an efficient link with simulation tools. An interesting perspective will be the development of automated interfaces between the architecture models and a simulation environment. Another interesting perspective is the integration of the security aspect into

the SoS mission process to identify vulnerabilities at an early stage as proposed in Ref. 48.

ORCID

Imane Cherfa https://orcid.org/0000-0003-3972-2353
Nicolas Belloir https://orcid.org/0000-0002-0163-8757

REFERENCES

1. Estefan JA. *Survey of Model-Based Systems Engineering (MBSE) Methodologies*. INCOSE; 2008.
2. Woodcock J, Larsen PG, Bicarregui J, Fitzgerald J. Formal methods: practice and experience. *ACM Comput Surv*. October 2009;41(4):1-36.
3. Council on Systems Engineering I. *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. Hoboken, NJ: Willey; 2015.
4. Nielsen CB, Larsen PG, Fitzgerald J, Woodcock J, Peleska J. Systems of systems engineering: Basic concepts, model-based techniques, and research directions. *ACM Comput Surv*. September 2015;48(2):18:1-18:41.
5. Vesonder G, Verma D. *RT-171: Mission Engineering Competencies*. Technical Report; 2018.
6. Sousa-Poza A. Mission engineering. *International Journal of System of Systems Engineering*. 2015;6:161.
7. Sheehan JH, Deitz PH, Bray BE, Harris BA, Wong ABH. *The Military Missions and Means Framework*. U.S. Army Materiel Systems Analysis Activity; 2004.
8. Military Agency for Standardization. *STANAG: Formats for Orders and Designation of Timings, Locations and Boundaries*, 9th ed. NATO; 2014.
9. Jamshidi M. System of systems engineering—New challenges for the 21st century. *IEEE Aerosp Electron Syst Mag*. May 2008;23(5):4-19.
10. Sage AP, Cuppan CD. On the systems engineering and management of systems of systems and federations of systems. *Inf Knowl Syst Manage*. December 2001;2(4):325-345.
11. Maier MW. Architecting principles for systems-of-systems. *Syst Eng*. 1998;1(4):267-284.
12. Dahmann J, Rebovich G, Lane J, Lowry R, Baldwin K. An implementers' view of systems engineering for systems of systems. *IEEE International Systems Conference*, Montreal, Canada; 2011:212-217.
13. Lowe PN, Chen MW. System of systems complexity: Modeling and simulation issues. *Proceedings of the Summer Computer Simulation Conference, SCSC '08*, Vista, CA: Society for Modeling; Simulation International; 2008:36:1-36:10.
14. Cole R. The changing role of requirements and architecture in systems engineering. *IEEE/SMC International Conference on System of Systems Engineering*, Los Angeles, CA; 2006:6-10.
15. Lane JA, Dahmann JS. Process evolution to support system of systems engineering. *Proceedings of the 2nd International Workshop on Ultra-large-scale Software-intensive Systems*, ULSSIS '08, New York, USA: ACM; 2008:11-14.
16. Lykins H, Friedenthal S, Meilich A. Adapting UML for an object oriented systems engineering method (OOSEM). *INCOSE International Symposium*, Vol. 10, 2000:490-490497.
17. Friedenthal S, Moore A, Steiner R. *A Practical Guide to SysML: The Systems Modeling Language*. 3rd ed. San Francisco, CA: Morgan Kaufmann Publishers Inc.; 2014.
18. Douglass BP. White paper: The Harmony Process. 2005.
19. Department of Defense. *Systems Engineering Guide for Systems of Systems*. 2008.
20. Dombkins D. *Complex Project Management: Seminal Essays/by David H. Dombkins*. North Charleston, SC: BookSurge Publishing; 2007.
21. Hans-Peter H. White paper: SysML-Based Systems Engineering using a Model-Driven Development Approach. 2008.
22. Morkevicius A, Aleksandraviciene A, Mazeika D, Bisikirskiene L, Strolia Z. MBSE Grid: A simplified SysML-based approach for modeling complex systems. *INCOSE International Symposium*, 2017;27(1):136-150.
23. Murray C. White paper: Rational Unified Process for Systems Engineering, RUP SE, Version 2.0. 2003.
24. Ingham MD, Rasmussen RD, Bennett MB, Moncada AC. Engineering complex embedded systems with State Analysis and the Mission Data System. *AIAA J Aerosp Comput Inform Commun*. 2005;2:507-536.
25. Department of Defense. *Defense Acquisition Guidebook*. Washington, DC: U.S. Dept. of Defense, Pentagon; 2010.
26. Dahmann J, Rebovich G, Lane JA, Lowry R. *System engineering artifacts for SoS*. 2010 IEEE International Systems Conference, San Diego, CA, USA; 2010:13-17.
27. U S Department of Defense. *DoDAF Architecture Framework Version 2.02*. 2010. Available at: https://dodcio.defense.gov/library/dod-architecture-framework/.
28. UK Ministry of Defence. *MOD Architecture Framework (MODAF)*. 2004. Available at: https://www.gov.uk/guidance/mod-architecture-framework.
29. Information Technology - Object Management Group. Unified Profile for DoDAF and MODAF (UPDM), 2.1.1. 2017.
30. Lock R, Sommerville I. Modelling and analysis of socio-technical system of systems. *Proceedings of the 15th IEEE International Conference on Engineering of Complex Computer Systems*, ICECCS '10, IEEE Computer Society, Oxford, UK; 2010:224-232.
31. Lochow T, Sanduka I, Bullinga R, et al. Concept Alignment Example description. 2013.
32. COMPASS Consortium. *The Compass Project*. 2014. Available at: http://www.compass-research.eu/.
33. Silva E, Cavalcante E, Batista T. Refining missions to architectures in software-intensive systems-of-systems. *IEEE/ACM Joint 5th International Workshop on Software Engineering for Systems-of-Systems and 11th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems (JSOS)*, Buenos Aires, Argentina; 2017:2-8.
34. Silva E, Batista T, Oquendo F. A mission-oriented approach for designing system-of-systems. *10th System of Systems Engineering Conference*, SoSE 2015, San Antonio,TX, USA; 2015:346-351.
35. Luzeaux D, Ruault JR. *Systems of Systems*. ISTE Ltd; 2010.
36. Cherfa I, Sadou S, Belloir N, Fleurquin R, Bennouar D. Involving the application domain expert in the construction of systems of systems. *13th Annual Conference on System of Systems Engineering (SoSE)*, Paris, France; 2018:335-342.
37. Bresciani P, Perini A, Giorgini P, Giunchiglia F, Mylopoulos J. Tropos: An agent-oriented software development methodology. *Auton Agent Multi Agent Syst*. May 2004;8(3):203-236.
38. Van Lamsweerde A. Requirements engineering: From craft to discipline. *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, New York, USA: ACM; 2008:238-249.
39. Object Management Group. *Systems Modeling Language V1.5*. 2017. Available at: http://www.omg.org/spec/SysML/1.5/.
40. Office of the Under Secretary of Defense for Acquisition Technology and Logistics. *Risk Management Guide for DOD Acquisition*. 6th ed. (Version 1.0). Washington, DC: Defense Technical Information Center; 2006.
41. International Organization for Standardization. *ISO 31000 Risk Management-Guidelines*. 2nd ed. 2018.
42. dos Santos Soares M, Vrancken J, Verbraeck A. User requirements modeling and analysis of software-intensive systems. *J Syst Softw*. 2011;84(2):328-339.
43. Gorod A, White BE, Ireland V, Gandhi SJ, Sauser B. *Case Studies in System of Systems, Enterprise Systems, and Complex Systems Engineering*. Complex and Enterprise Systems Engineering. 1st ed. Boca Raton, FL: CRC Press; 2014.

44. Groupement des Industries de Défense et de Sécurité terrestres et aéroterrestres. Gestion des foules. GICAT; 2018.
45. FIFA. *Stadium Safety and Security Regulations*. FIFA; 2018.
46. Dahmann J, Markina-Khusid A, Doren A, Wheeler T, Cotter M, Kelley M. SysML executable systems of system architecture definition: A working example. *Annual IEEE International Systems Conference (SysCon)*, Montreal, Quebec, Canada; 2017:1-6.
47. Benabidallah R, Sadou S, Ahmed-Nacer M. Using system of systems' states for identifying emergent misbehaviors. *27th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises*, WETICE 2018, Paris, France, June 27-29, IEEE Computer Society; 2018:66-71.
48. Messe N, Belloir N, Chiprianov V, Cherfa I, Fleurquin R, Sadou S. Development of secure system of systems needing a rapid deployment. *14th Annual Conference System of Systems Engineering*, SoSE 2019, Anchorage, AK, USA, May 19-22, 2019:152-157.

AUTHOR BIOGRAPHIES

IMANE CHERFA is a PhD student at Blida1 University, Algeria and the University of South Brittany, France (Cotutelle). She is a permanent Lecturer-researcher at the Computer Science Department of Blida1 University since 2012, and permanent member of the LRDSI labs (Blida1 Computer Science Laboratory) since 2018. She received the BSc Eng. Degree (2008) and the Magister Degree (2012) from Blida1 University where she worked as Part-time lecturer from 2009 to 2012. Her research interests include Systems-of-Systems modeling and mission engineering.

NICOLAS BELLOIR is an associate professor at the Military Academy of St-Cyr Coetquidan (St Cyr), France, since 2016. From September 2005 to May 2017, he was an associate professor at the University of Pau (UPPA), France. Former member of the LIUPPA (UPPA Computer Science Laboratory) from May 2001 to August 2016, he is permanent member of the CREC (Research Center of St-Cyr) and of the IRISA labs (Archware team) since 2016. He received a master degree of the University Paul Sabatier (Toulouse) in 1999. He worked as real-time and embedded systems engineer for several companies from 1999 to 2001. He received the PhD from the UPPA in December 2004. His research areas include software and system engineering, requirement engineering, semiformal design languages (UML, SysML, DSML), and model-based engineering. Currently he is focusing on cybersecurity of Systems-of-Systems.

SALAH SADOU is a Professor of Computer Science at the University of South Brittany. He received the BSc Eng. Degree (1987) from Algiers University of Science and Technology, and the MSc (1988), PhD (1992) from Ecole Centrale de Lyon, and HDR (Research Direction Habilitation) degree in Computer Science (2003) from the University of South Brittany. His research interests are centred on languages, processes, and tools for designing and engineering systems where the evolution acts as a first-class entity. His past research interests included architectural description languages with nonfunctional properties as first-class entities, software restructuring (from object-oriented to component-oriented), component-based description languages, and software quality. His current research work concern systems of systems construction with a focus on the security aspect.

RÉGIS FLEURQUIN is an associate professor of Computer Science at the University of South Brittany, France. He is a permanent member of the IRISA labs (Archware team). He received a master degree of INSA Lyon (National Institute of Applied Sciences, France) in 1991, a PhD from INSA Toulouse in 1996 and HDR (Research Direction Habilitation) degree in Computer Science from the University of South Brittany in 2010. His research interests are centered on software architecture, software quality, and model-based engineering. Currently, he is focusing on cybersecurity of Systems-of-Systems.

DJAMAL BENNOUAR is a Professor at Bouira University, Algeria, and the Director of the LIMPAF laboratory (Software System and Sensor Networks for Agriculture and Forestry). He obtained the Magister Degree from the National Institute for Computer Science (INI), Algeria, in 1993 and the PhD degree from the Ecole Superieure d'Informatique (ESI), Algeria, in 2009. He conducted various research related to VLSI CAD Frameworks (HDL, Inter tools communication, Engineering Databases), Computer Networking and Software Product Lines for E-Government. Currently, his main research interests include Software Architecture, Software Product Lines, and Automatic evaluation of answers to open ended question. He is supervising a number of PhD students preparing their thesis in Software Architecture, Software Architecture Approach for System On Chip, Software Product Lines, and Automatic Student Assessment.

How to cite this article: Cherfa I, Belloir N, Sadou S, Fleurquin R, Bennouar D. Systems of systems: From mission definition to architecture description. *Systems Engineering*. 2019;1–18. https://doi.org/10.1002/sys.21523

2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)

Asset-Oriented Threat Modeling

Nan Messe†, Vanea Chiprianov*, Nicolas Belloir*, Jamal El-Hachem*, Régis Fleurquin*, Salah Sadou*
* *Archware - IRISA, France*
† *DiverSE, IRISA, France*
Email: firstname.lastname@irisa.fr

***Abstract*—Threat modeling is recognized as one of the most important activities in software security. It helps to address security issues in software development. Several threat modeling processes are widely used in the industry such as the one of Microsoft SDL. In threat modeling, it is essential to first identify assets before enumerating threats, in order to diagnose the threat targets and spot the protection mechanisms. Asset identification and threat enumeration are collaborative activities involving many actors such as security experts and software architects. These activities are traditionally carried out in brainstorming sessions. Due to the lack of guidance, the lack of a sufficiently formalized process, the high dependence on actors' knowledge, and the variety of actors' background, these actors often have difficulties collaborating with each other. Brainstorming sessions are thus often conducted sub-optimally and require significant effort. To address this problem, we aim at structuring the asset identification phase by proposing a systematic asset identification process, which is based on a reference model. This process structures and identifies relevant assets, facilitating the threat enumeration during brainstorming. We illustrate the proposed process with a case study and show the usefulness of our process in supporting threat enumeration and improving existing threat modeling processes such as the Microsoft SDL one.**

***Index Terms*—threat modeling (process), asset-based reference model, asset identification, attack pattern**

I. Introduction

Threat modeling is recognized as one of the most important activities in software security [18]. It aims at identifying a coverage of all possible threats [4] and preventing and/or mitigating the effects of threats and attacks on a software system. Several threat modeling methods exist, reviewed for example in [35], [38]. As part of all these methods, threat enumeration is at its core [6], which is traditionally carried out in brainstorming sessions. Current widespread threat modeling methods (such as STRIDE [17], OCTAVE [2], PASTA [36], etc.) are coarse-grained and require in-depth security knowledge. There is no detailed description of a procedure to support the brainstorming sessions, and no reference model to be used by such a procedure [16], [29]. Due to the lack of guidance, the lack of sufficiently formalized process, the high dependence on actors' knowledge and the variety of actors' background, these sessions are often conducted sub-optimally and require significant effort [9].

Thus, several research challenges have been recently identified [16], such as 1) developing a reference model, which makes it possible to share threat modeling artifacts in a standardized manner for the reuse, education, and benchmark and 2) defining a process that better supports the interactions among threat modeling participants, consequently, allowing a better knowledge reuse across projects, experts, and organizational boundaries. To improve current threat modeling processes and rise to the identified research challenges, we propose an asset identification process, to help participants collaboratively identify assets, which are significant for both business stakeholders and product team members, as well as for the security experts. This structured process employs a number of concepts and relations, which we organise into a reference model. Moreover, to increase the knowledge reuse degree and reduce the reliance on subjective experience, we propose to construct a vulnerable asset library as part of a threat library.

The paper is structured as follows: Section II presents the background of threat modeling processes and the motivation of this paper. In Section III, we present our approach, including the need of reworking on the *asset* concept, structuring the threat modeling knowledge into an asset-based reference model and defining an asset identification process. In Section IV, we show the application of our approach, including a library of vulnerable assets, which we extract from common security knowledge bases such as CAPEC by applying several heuristic rules. We also illustrate in this section a case study by firstly applying the Microsoft SDL threat modeling process, and then integrating our asset identification process into it to improve its results. Then we discuss the advantages and limitations of our approach. Related works are discussed in Section V. Finally, we conclude the paper in Section VI.

II. Background and Motivation

In this section we first define what the threat modeling is and show its importance in secure software development. Then we make an inventory of the current threat modeling processes and highlight their limitations. Finally, we identify several needs or requirements that remain to be satisfied, which point out our motivation.

A. Threat modeling definition

There are numerous definitions of threat modeling in literature, used in different and perhaps incompatible ways [38]. For our purpose, we define the threat modeling as a systematic process of identifying and analyzing threats (i.e. potential attacks), which involves the understanding of threat agents goals and adversaries actions in attacking a system, based on that system's assets [37].

Threat modeling can occur at any time during the software development lifecycle, but it's more efficient to be

978-1-6654-0392-4/20/$31.00 ©2020 IEEE
DOI 10.1109/TrustCom50675.2020.00073

performed during the early requirement and the architecture/design phases [14], because fixing an issue that involves reworking on a conceptual model rather than significant re-engineering can save cost, development time and protect the system from high impact attacks [34], [37].

Threat modeling process is also a collaborative process where participants include: business stakeholders; product-team members from all product development phases, such as enterprise, software and application architects, development leads, IT infrastructure specialists, engineers; security experts such as security analysts, security architects, threat modeling experts [32], [34].

Threat modeling is important because it helps in 1) identifying business-logic flaws and other critical vulnerabilities that expose core business assets, 2) enriching assessments with new potential attack vectors, 3) prioritizing the types of attacks to address 4) mitigating the risks more effectively and 5) fixing issues early in the development process [32], [35].

B. Threat modeling process

Several threat modeling processes including various activities are proposed in literature [4], [15] and widely used in industry (Microsoft [26], [28], [34], CIGITAL [32] and EMC [6]), as shown in Table I. We summarize the threat modeling process into four main phases:

1) **Asset Identification phase**: It is centered on identifying security goals, modeling domains (by characterizing the system, usually by decomposing it and describing its components and data flows using (annotated) architecture/design diagrams) and identifying valuable assets.

2) **Threat Enumeration phase**: It is focused on identifying threats, together with attackers (their motivation and skill) and vulnerabilities, and enumerating and documenting resulted threats. This phase is often conducted in brainstorming meetings, sometimes guided by a threat library.

3) **Threat Prioritization phase**: It is based on the result of threat enumeration, to rate threats and assess risks. This phase can be either considered as an internal or external activity [35].

4) **Mitigation phase**: It aims at resolving threats by proposing security mitigations and by verifying them.

It is worth noting that not all the current approaches in Table I include the asset identification phase, which is nonetheless an important step. The activity of identifying asset is a bridge between domain modeling and threat identification, however, only a few works address this activity. Identifying assets is essential because it takes both into account the modeling of the domain under consideration, and the will of stakeholders to protect valuable elements. Without this step, the later threat enumeration and prioritization phases would be less efficient. Some approaches mention the activity of "identifying asset" [4], [15], however no detailed guidance or formalized process about how to systematically conduct this activity is proposed.

As a prerequisite for the threat enumeration (which is at the core of the threat modeling process [6], [16]), the quality of the asset identification impacts directly the threat enumeration and indirectly later phases. Therefore, the early phases (asset identification and threat enumeration) of threat modeling are crucial for the success of threat modeling. However, the activities in these phases are often conducted in brainstorming sessions [29], the results of which depend highly on the human expertise, experiences and collaboration, even with the support of such methods as STRIDE (a coarse-grained guiding method used largely in industry).

Conducting threat modeling thus requires ideally a sound knowledge of a system's technical domain and sufficient security expertise to consider both generic and specific attacks for various specific contexts [37]. With the threat modeling process going on, more and more security expertise is required in each phase. However, the need of security knowledge can "leave most 'off-the-street' developers estranged" [37], with the result that threat modeling is performed sub-optimally or with significant effort involved, or not performed at all. Even after security training, the threat modeling process is still difficult to execute [32].

C. Motivation

Threat modeling is still at a low level of maturity [16] and several key research challenges and/or requirements have been identified:

1) It is important to hold a successful brainstorming meeting [29], which is still a subjective and unstructured activity. It should follow a methodical approach in enumerating threats, while still letting participants think about the problem creatively. It thus needs a guidance that is more prescriptive, formal, reusable and less dependent on the aptitudes and knowledge of the participants. Meanwhile, the cause of the high number of overlooked threats is also worthy of investigating [26].

2) The current threat modeling processes require a certain security knowledge level, making it a non-trivial task for participants with limited security knowledge. Proposed widespread threat modeling methods (such as STRIDE [17], OCTAVE [2], PASTA [36], etc.) are abstract, coarse-grained and require in-depth security knowledge. Wrong decisions are thus made based on insufficient knowledge about the security domain, threats, possible countermeasures and the own infrastructure. An in-depth reason is that security terminology is vaguely defined. This leads to confusion among experts as well as the people who should be counseled and served [7]. There is thus a need to propose a method that can be easily used or understandable by security novices.

3) Moreover, a successful communication among threat modeling participants requires that they share their knowledge and points of view with as little bias and as few misunderstandings or confusion as possible [9]. Without a shared terminology communication, especially in a complicated domain like security, threat modeling cannot be successful [8]. Incorrect results could thus be caused by the misinterpretation of some template threats in the checklists. Therefore, there is a need of a common language or a common concept that can be understood by all participants.

Phase	Asset Identification			Threat Enumeration				Threat Prioritization		Mitigation	
Activity / Paper	Identify security goal	Model domain	Identify asset	Identify threat	Enumerate &document threat	Describe attacker	Identify vulnerability	Rate threat	Assess risk	Mitigation	Verification
Torr (2005) [34]		x		x						x	x
Shostack (2008) [28]		x			x					x	x
Scandariato (2013) [26]		x		x	x						
Beckers (2013) [4]		x	x	x	x	x					
Dhillon (2011) [6]		x		x					x	x	
Steven (2010) [32]	x	x		x			x				
Kamatchi (2016) [15]		x	x	x	x			x			

TABLE I: An inventory of threat modeling processes

III. Structuring the asset identification phase

Our proposal addresses the 3 preceding needs. It aims to structure the asset identification phase: to associate it with a well-defined process promoting the manipulation of a set of precise and well-structured concepts and exploiting, if needed, a security knowledge base to limit the negative impact of the lack of experience in security. These elements can be used to guide the actors during the brainstorming sessions. This proposal is based on a novel refinement of the concept of *asset* which we describe at first. Secondly, we show how we use this refinement to structure the universe of threat modeling knowledge using a reference model. This reference model is used to structure the common language of the information handled by all participants during this phase. But also it can be used as a language with which one can capitalize in a knowledge base the state of the art in security. We next present a process based on this reference model to lead the asset identification phase.

A. Reworking on the asset concept

Our proposal is based on the *asset* concept. This concept can be easily understood by business stakeholders and by product team members [19], [24]. It is naturally well-known by security experts. It can therefore act as a shared concept between all participants in the threat modeling process. It also provides a solid common base for further activities, especially threat enumeration.

There are numerous definitions of the *asset* concept in literature. For example, ISO 21827 [1] defines *asset* as anything that has value to the organisation, such as data, hardware, software or networks. Similar definitions focusing on the value of an asset, which can be subjective, commercial, and vary in a wide range, are presented in [25]. Several definitions look at assets from the attackers' point of view, defining them as the things that an attacker tries to steal, modify, or disrupt, and considering their relations with threats [34]. Other definitions consider the relations of assets with vulnerabilities/exposures/weaknesses and countermeasures [22].

These definitions are too generic, too abstract and too wide-encompassing; entities that have various natures can be included in these definitions. Such multiple overlapping definitions, including things attackers want, things stakeholders are protecting, and stepping stones, can "trip you up" [29]. It may thus lead to misunderstandings and confusion among the threat modeling participants.

We therefore consider two viewpoints for the *asset* concept:

The domain expert's viewpoint: We consider as *domain experts*, participants in the threat modeling process who are not security experts, such as: business stakeholders and product-team members from all product development phases, such as enterprise, software and application architects, development leads, IT infrastructure specialists and engineers. They deal with **Domain Asset (DA):** Anything that has value for them, towards the fulfilment of the function and goal of the system, together with the assurance of its properties. DA is artifact of a particular system architecture.

The security expert's viewpoint: We consider as *security experts*, participants in the threat modeling process who have sufficient security knowledge, such as: security analysts, security architects and threat modeling experts. They deal with **Vulnerable Asset (VA):** Anything that has value for them. It has vulnerabilities that can be menaced by threats. Hence it is the direct, core target of the attacker. If it is compromised, it can impact relevant domain asset (DA). Therefore, they need protection to reduce threats and prevent attacks. VA is system artifact appearing in more or less abstract attack patterns or vulnerability bases.

VA and DA are all system artifacts but appear in a different context: respectively in an attack pattern description and in a system architecture model. They can also have different abstraction levels. However, they may also include elements which are exclusive to any one of them. As such, the domain assets may include assets which may not be vulnerable, and therefore not identified as vulnerable assets. Similarly, vulnerable assets may include assets which are not used in the domain models. Determining the common assets between the domain and vulnerable assets is not trivial. However, it is essential, because they represent the domain assets that are also vulnerable and therefore need protection.

The domain assets that are also vulnerable assets constitute therefore a new type of asset, understandable by both the domain and security experts. We call this new type of asset **Vulnerable Domain Asset** (VDA). The VDA is therefore anything that has value for the domain expert, but also has vulnerabilities that can be menaced by threats. As it is a domain asset, it is also domain specific. Our VDA concept has precursors in the literature. For example, [29] remarks that the most common usage of asset in discussing threat models seems to be a marriage of "things attackers want" and "things you want to protect". However, in previous works, this idea is not further developed to show how to differentiate and organize different types of assets.

B. Structuring the threat modeling knowledge

Now that we have refined the *asset* concept, we propose to structure the universe of information manipulated during brain-

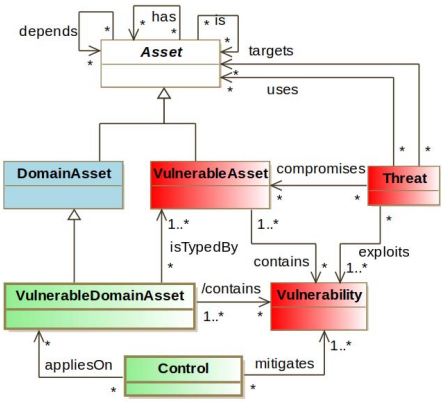


Fig. 1: Asset-based reference model

storming using a *reference model*. The definition of this model has two objectives: 1) fixing and structuring the discourse during brainstorming sessions and 2) allowing the capturing of security knowledge from the literature in a form which can then be reused during brainstorming. This reference model is presented in Figure 1. *Asset* is the core concept of our model. It is an abstract class. As discussed above, we specialize the concept of *Asset* into *DomainAsset* (DA) and *VulnerableAsset* (VA). The *VulnerableDomainAsset* concept (VDA) is a type of both *DomainAsset* and *VulnerableAsset*. Both *VulnerableAssets* and *VDA* can *have Vulnerabilities*, which can be *exploited* by *Threats*. Thus *Threats* can *compromise* VA and thus VDA. In its compromise actions, a *Threat* may *target* an *Asset* (both *Domain* and *Vulnerable*) *using* other compromised *Assets* in the process. To mitigate the *Vulnerabilities*, *Controls* can be *applied* on *VDA*.

Each *Asset* can have three relationships with other Assets: *is*, *has* and *depends*. The *is* relation captures the generalisation between *Assets* of different abstraction levels. It captures an iterative refinement of assets. For example, the domain experts define the list of domain assets coming from the domain architecture model. During design, they can progressively refine this list from more abstract assets to more concrete ones. Similarly, the security experts define an hierarchy from more abstract (coming from abstract attack pattern) to more concrete vulnerable assets. Moreover, an *Asset* may be composed of other *Assets*. We model this through the *has* relation. We also introduce the *dependency* relation between *Assets*. A dependency exists between two elements if changes to one element (the supplier) may cause changes to the other (the client) [11].

These three relationships (generalisation, composition and dependency) are very common in system architecture modeling. It is worth noting that this kind of modeling promotes a data structure similar to that of a B-tree [3], even if other data structures are also possible, such as class diagram. We choose B-Tree structure because it can be easily coupled with and extended from Attack Tree, as we deal with security aspect. A B-tree is a tree data structure where each level can have one or more *children* nodes. Each node may be thought of as a kind of list, containing several entities called *keys* (related to the origin of B-trees for databases). In our case, the *Assets* related by an *is*, are similar to the *children* nodes of a B-tree. For example, in Figure 2, VA2 and VA6 are *children* of VA1, and VA3 is a *child* of VA2. The *Assets* related by *has* and *depends*, correspond to the *keys* of a B-tree. VA4 and VA5 are keys related to VA2, related respectively by *has* and *depends* relations. In our data structure, we just take inspiration form the idea of B-trees, but are not interested in their properties, such as self-balancing. We choose B-Tree structure because it allows to show all the three relations of different dimensions inside one tree. More precisely, B-Tree allows showing the generalization relation vertically, and showing composition and dependence relations horizontally inside each child nodes, to align with the relations of different dimensions among different assets. This similarity may also enable a higher degree of automation for the asset identification process in the future. Moreover, this is close to the structure of existing security knowledge bases, such as CAPEC, facilitating extraction from them (cf. Section IV-A).

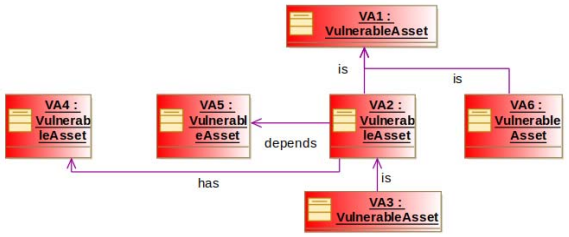


Fig. 2: Vulnerable asset B-tree

To reduce the level of security expertise required, threat modeling can be supported by threat libraries (structured or unstructured lists of threats), which have been found particularly effective in industry scenarios [37]. However, non-security experts, such as domain experts, have to be trained to better use threat libraries, as they require a minimum security knowledge to understand security jargon. Therefore, we think that it is useful to construct a vulnerable asset library, which can enrich the threat library. To help the asset identification process, we thus propose to construct a library of vulnerable assets. The VA library aims to classify a wide variety of abstract, system- and technique-independent VA, which keeps the asset identification and threat enumeration manageable, increases the VA library's applicability and reusability, and makes it both more practical and more useful for security novices and experts alike. For the library to be well integrated with the asset identification process, we propose that the library and the reference model presented follow the same structure for the VA. Part of construction process of the VA library is presented in Section IV-A.

C. Defining the asset identification process

After refining the *Asset* concept and proposing an asset-based reference model to structure the knowledge, we present a process to help actors in the identification of threats targeting the assets. This process is shown in Figure 4 and the general view is summarized in Figure 3. This process can be launched

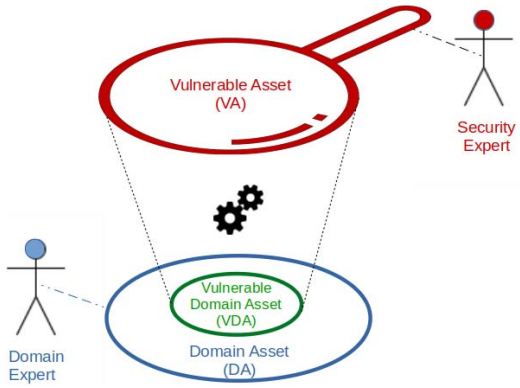


Fig. 3: General view of asset identification

regardless of the software development stage and therefore on more or less abstract models.

On one hand, DA, obtained from domain models such as enterprise, system and software architectures, is structured by relationships such as generalization, composition and dependency. On the other hand, security experts identify Vulnerable Assets (VA) relevant to the types of elements present in the model being designed. This list of VA can be populated from the security experts' knowledge, as well as be extracted from common security knowledge databases, thus promoting reuse. This extraction is a non-trivial process, involving threat libraries, attack patterns (e.g. CAPEC), attack trees, vulnerabilities, etc. Thus, we promote the setting up a VA library synthesizing current knowledge of the field in a format that respects our reference model.

Since DA and VA are similarly structured by the *is*, *has* and *depends* relationships defined in the reference model, the goal of the asset identification process is to bridge the gap between these two sets of assets (cf. Figure 3) and identify VDA (i.e. Domain Assets which are also Vulnerable). The security experts project or instantiate these VA on the DA. A comparison is made by actors to identify if a mapping occurs between VA and DA. If mappings are identified, they represent the VDA. It is therefore noteworthy that the matching process instantiates *abstract* VA into *concrete* VDA. Discovering the VDA further enables identifying security mitigations (i.e. controls), based on their vulnerabilities. In this way, our approach uses domain-independent, general, security threat and attack knowledge to identify and protect domain-specific VDA.

Figure 4 illustrates a fine-grained asset identification process. It takes as inputs a DA list and a VA tree. The DA list is a result of domain experts identifying assets specific to their domain. The VA tree results from the security experts using their knowledge to identify generic vulnerable assets, and it can be enriched with information extracted from security knowledge bases or a VA library.

The asset identification process can traverse the vulnerable asset tree (respecting to B-tree) either in a depth-first or in a breadth-first manner. In this paper, we choose to present a breadth-first strategy. As such, the process *selects* the VA tree children situated at the current vertical "i" level (i.e. all the VA linked through an *is* relation to the VA of the previous parent level). For instance, when considering the example in Figure 2, concerning level "i=1", the children are VA2 and VA6. For each VA child, the domain and security experts *compare* its syntactic and semantic similarity with each DA in the current domain asset list. If a VA_k, from the list of VA level i children, is found similar with a DA_j, from the DA list, further similarities are searched. For this, the VA tree is traversed horizontally, and the keys attached to that VA_k are *selected* (i.e. the VA linked through *has* and *depends* relations). Let us suppose that for the Figure 2 example, VA2 is found similar with a DA, then its key list containing VA4 and VA5 is *selected*.

The domain experts select among VA_k keys those which are involved in the domain. VA_k keys that are involved in the domain, discovered at this "i" iteration, are added to the current DA list, enriching the DA list for the next iteration. As they are initially VA, but also in the domain, they are actually VDA, and therefore are also added to the VDA list. Let us suppose that in our example VA4 is a key that is involved in the domain. At the end of the "i" iteration, the DA list additionally contains VA4 and the VDA list contains VA2 and VA4. Then, if no more VA_k key is found being involved in the domain, the process advances to the next VA tree level (i.e. "i=i+1"), until there are no more levels (i.e. "i==n"). The DA and VDA list are enriched with the iteration of each "i" level.

Once the VDA list has been enriched, it is used as a bridge towards threat enumeration. Security and domain experts may use it to propose security mitigations to the identified vulnerabilities.

IV. Case Study

In this section we want to highlight the gain obtained through the use of our approach. To do this, we proceed as follows: we first present a process of constructing a vulnerable asset library by leveraging CAPEC and respecting the B-tree structure of the reference model, in order to show the possibility of knowledge reuse. Secondly, we illustrate a case study by firstly applying the Microsoft SDL threat modeling process, and then integrating our asset identification process into it to improve its results. We believe that the integration of our asset identification process into the Microsoft process as a complementary step can improve the detection of more relevant threats. The advantages and limitations of the resulting enriched process, compared to those of the sole Microsoft process, are discussed at the end of this section.

A. The process of constructing a vulnerable asset library by leveraging CAPEC

As we mentioned before, Vulnerable Asset (VA) represents security experts' viewpoint and it is domain-independent. Therefore, VA can be extracted from existing security knowledge bases for the reuse in different contexts or domains. In this section, we stress the importance and reusability of this extraction and present part of extraction rules by leveraging

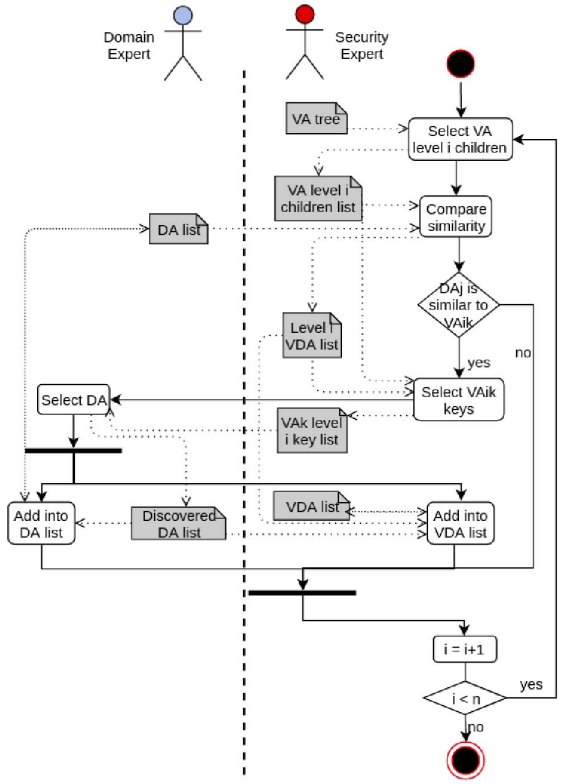


Fig. 4: Asset identification process

well-known attack pattern knowledge bases such as CAPEC and by respecting the B-tree structure.

Attacks are possible realisations of threats [31]. Therefore attack descriptions can be useful in enumerating threats. To construct this library, we can leverage existing attack databases such as CAPEC [20], OWASP [10] and ATT&CK [21]. However, they are defined in natural language and possibly ambiguous, which make them difficult to be processed automatically. At the current state of advancement, we identify a number of heuristic rules which can be enriched in the future. These rules help partially extract VA and relations between these VA that can be compromised by threats.

We show these extraction rules by leveraging CAPEC, which is one of the most popular and structured attack databases. In CAPEC, the attacks belong to different levels of abstraction: *view, category, meta, standard and detailed*. We focus on the *meta, standard and detailed* abstraction levels, because *view* and *category* levels are too abstract to be reused effectively. 1) A *meta* attack pattern is "an abstract characterization of a specific methodology or technique used in an attack", and "a generalization of related group of *standard* attack patterns". 2) A *standard* attack pattern is "focused on a specific methodology or technique used in an attack". 3) A *detailed* attack pattern "provides a low level of detail, typically leveraging a specific technique and targeting a specific technology, and expresses a complete execution flow". *Detailed* attack patterns are more specific than *meta* and *standard* attack patterns. The links between these abstraction levels are modeled through "*childOf*/*parentOf*" relations. This hierarchical attack/threat structure can help us identify *is* and *has* relations between VA. Moreover, there are also relations of "*canFollow*/*canPrecede*" between attacks/threats in CAPEC, which can help us identify *depends* relation between VA.

Based on CAPEC attack natural language descriptions, we define several VA extraction rules:

1) If the name of attack pattern contains the keyword "contaminate", or "poison", or "leverage", or "manipulate", or "abuse", or "exploit" or "misuse", then the noun set after any of these keywords is selected as a vulnerable asset (VA). For example, for the *detailed* attack pattern "Poison web service registry" (CAPEC-51), the "web service registry" is a vulnerable asset;

2) If the name of attack pattern contains the keyword "manipulation", or "poisoning", or "tampering" or "alteration", then the noun set before any of these keywords is extracted as a vulnerable asset (VA). For example, for the *standard* attack pattern "Web service protocol manipulation" (CAPEC-278), the "web service protocol" is a vulnerable asset;

3) If the name of attack pattern contains the keyword "injection", or "inclusion" or "insertion", then the noun set before any of these keywords is selected and we add the literal "Untested" before and "Input" after this noun set, the whole literal word is considered as a VA. For example, for the *standard* attack pattern "XML injection" (CAPEC-250), "XML" is selected and added by the above prefix and suffix. As a result, "UntestedXMLInput" is a vulnerable asset.

There are three possible relations (*is, has, depends*) between VA, as mentioned in Section III-B. By leveraging CAPEC, we can also extract the relations between VA.

4) The "childOf" relation between two attack patterns is translated into either "*is*" or "*has*" relation between two corresponding VA, because "ChildOf" in CAPEC can present either a specialisation or a decomposition relation. For example, on one hand, the "SOAP" VA extracted from the *detailed* attack pattern "SOAP Manipulation" (CAPEC ID 279), *is* a type of "Web Services protocol" VA. On the other hand, the "XML" VA *has* "DTD", "XPath" and "XQuery" VA, extracted respectively from three *detailed* attacks (CAPEC IDs respectively 228, 83, 84). Therefore, the reasoning about the decision comes from the security experts who extract VA;

5) The "canFollow" relation between two attack patterns is translated into *depends* relation between two relevant VA, because if asset A_a is compromised by an attack/threat T_a, then a threat T_b, which can follow T_a, can compromise asset A_b, therefore asset A_b *depends* on asset A_a.

These rules allow us to extract VA from attack/threat patterns. For each extraction, the relation between the threat and the VA is stored. This allows to later find all the threats that compromise the same VA. In this way, our library contains the information about VA and threats that compromise them. At the current state of this paper, the VA extraction process is conducted manually by the first author. We believe that this extraction process can be implemented using techniques

such as parsing, text mining and/or bash/sh scripting to allow automation.

The VA library, as a part of threat library, lightens the dependency on attack knowledge. It aims to be utilized by both security and non-security experts. Therefore, the construction of the VA library can satisfy to the requirement 2 in Section II-C.

B. Illustration of the process integration

In this part, we first illustrate a case study using the Microsoft SDL threat modeling process to enumerate threats. As we will see, this process lacks of an "identifying asset" activity, which is a bridging step between the "domain modeling" and the "threat identification" activities. Therefore, we then illustrate the integration of our asset identification process into the Microsoft process as a complementary step to improve the detection of relevant threats.

1) Microsoft SDL threat modeling process illustration: Microsoft SDL threat modeling process is based on STRIDE, which is currently the most mature threat modeling method [27], and is implemented with the SDL threat modeling tool, which is available online [5].

There are four steps, described in [28], to conduct threat modeling process in Microsoft: 1) diagramming (by applying Data Flow Diagram - DFD), 2) threat enumeration (by applying STRIDE), 3) mitigation and 4) verification. As our paper aims to support the threat enumeration, we only present the first two steps. Microsoft implements the SDL threat modeling process into Microsoft threat modeling tool. This tool provides predefined DFD elements, as well as allowing users create new templates containing stencils (new elements) and threat types. We illustrate a case study with the tool.

As to the case study, we take the example of Web Sphere 7.0 application server, which is a software framework that hosts java-based web applications, allowing deploying and managing applications ranging from simple Web sites to powerful on-demand solutions. It is architected as a distributed computing platform that could be installed on multiple operating system instances, collectively referred to as a WebSphere cell. Its configuration information are tracked in XML configuration files throughout the cell.

The Microsoft process begins by characterizing the software or system (Web sphere 7.0 in our case), by decomposing it and describing its components and data flows, using DFD. There are four types of elements in DFD: external entity, process, data flow and data store. An excerpt of a possible decomposition of the Web Sphere 7.0 application server is presented in Figure 5, obtained using the Microsoft tool. It is decomposed into a process called "web service" and a data store termed "configuration file". "Web service" interacts with "configuration file" through a "general data flow". The above three DFD elements are predefined by the tool. Further DFD modeling of the case study is not presented here for the sake of readability and space reasons.

The next phase is the threat enumeration, which is conducted in a brainstorming meeting guided by "STRIDE by elements", supported by the threat list generated automatically by the tool. The threat list contains the threats that menace each DFD element or a group of DFD elements. As shown in Figure 5, the tool has found two threats concerning "Web Service", "Configuration File" and their interaction: 1) Spoofing of destination data store configuration file (belonging to the threat category of Spoofing) and 2) potential excessive resource consumption for web service or configuration file (Denial of Service).

In the brainstorming meeting, participants generally discuss and find out potential threats belonging to six threat categories of STRIDE (Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege) that can threaten the actual DFD element. At the current state of the work, we have not discussed with Microsoft SDL threat modeling experts. The quality and quantity of the results of the brainstorming meeting depends highly on participants. It is a highly subjective activity, the results of which are not reproducible. This makes it difficult for us to compare the brainstorming activity with our asset identification process. However, we believe that our process can help structure this activity, which we discuss in Section IV-C.

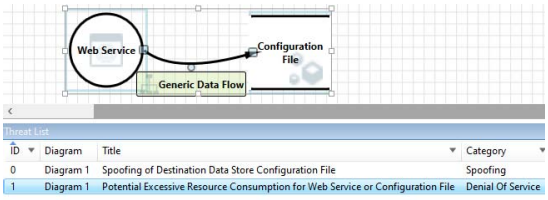


Fig. 5: Applying Microsoft SDL threat modeling tool

2) Integrating our process into Microsoft SDL threat modeling process: The Microsoft SDL threat modeling process begins by modeling the domain (by applying DFD), before identifying threats (by applying STRIDE). As we noted when discussing Table I, the Microsoft SDL process does not contain the activity of "identifying asset", which is a bridging step between "modeling domain" and "identify threats". Therefore, we present the integration of our asset identification process into the Microsoft SDL threat modeling process in the aim at discovering more relevant threats.

Based on the DFD model in Figure 5, we observe that there is a loss of information during the domain modeling: the "configuration document" is of the XML type. This information may be critical for threat enumeration. A reason for this loss of information is that XML document is not predefined by the tool.

Therefore, we add the "XML document" in the domain model for our asset identification process, based on the DFD modeled in Figure 5, in order to fill the gap between domain modeling and threat enumeration. We describe the domain model of the case study using UML class diagram. Other modeling languages can be used as well. The domain asset model is presented in Figure 6. Conforming to the description of "Web Sphere 7.0 application server", the "Configuration Document" is of type XML and is contained in the "Web Sphere Server", together with "Web Service".

To apply the asset identification process, on one hand, the domain experts produce the domain asset list, part of which is shown in Figure 6. The *WebSphere7.0* contains, among other components, a *ConfigurationDocument* and a *WebService*. These correspond to concrete (architecture) elements. The *ConfigurationDocument* can be generalized into an abstract (architecture) element *XMLDocument*.

On the other hand, the security experts use a vulnerable asset B-tree from the VA library (constructed using the heuristic rules presented in Section IV-A), part of which is presented in Figure 7. This vulnerable asset B-tree begins by a root called *VARoot* (VA_1), which is an artificial root to start the process and can be specialized by any of its children VA. In Figure 7, the VA UntestedCommandInput, UntestedCodeInput, UntestedXMLInput, UntestedSQLInput, UntestedDTDInput and UntestedXPATHInput are respectively extracted: from the *meta* attack patterns Command Injection (CAPEC-248) and Code Injection (CAPEC-242); from the *standard* attack patterns XML Injection (CAPEC-250) and SQL Injection (CAPEC-66); and from the *detailed* attack patterns DTD Injection (CAPEC-228) and XPATH Injection (CAPEC-83), respecting the rule 3 in Section IV-A. Other VA are omitted in this paper due to limited space.

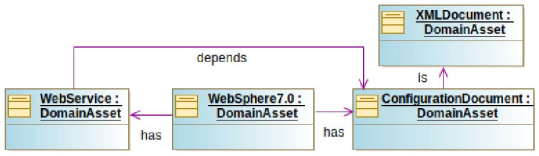


Fig. 6: An Excerpt of Domain Assets

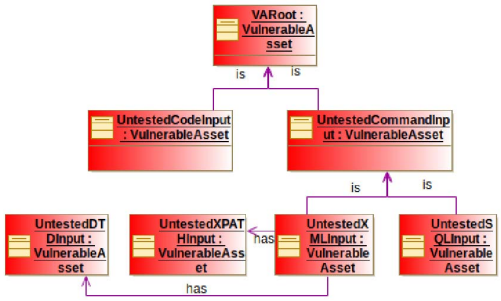


Fig. 7: An Excerpt of Vulnerable Asset Tree

We illustrate the asset identification process based on the VA tree in Figure 7. We initialize the process with i = 1, in this case, n is equal to 3. In the following, we illustrate each task of the process of Figure 4:

1) Select VA level i children: At the beginning of the process, i = 1, which is the *VARoot*. In our case, VA level 1 children are *UntestedCommandInput* (VA_{11}) and *UntestedCodeInput* (VA_{12});

2) Compare similarity: With the DA list provided by domain experts, security experts need to compare syntactical and semantic similarity between a vulnerable asset and a domain asset. Among the four domain assets *XMLDocument* (DA_1), *WebSphere7.0* (DA_2), *ConfigurationDocument* (DA_3) and *WebService* (DA_4), there is no similarity found when compared with *UntestededCommandInput* (VA_{11}) and *UntestedCodeInput* (VA_{12});

3) Therefore, for all DA_j, none of them is similar to VA_{1k}, which are children of VA level 1. In this case, i increments, now i is equal to 2, which is still lower than 3;

4) Select VA level i children: As no similarity is found from the upper level, the process advances to the lower level of the VA tree. For level i=2, there are two VA UntestedCommandInput and UntestedCodeInput, as shown in Figure 7. For the VA UntestedCommandInput, there are two children. Therefore, the VA level 2 children list contains UntestedXMLInput and UntestedSQLInput;

5) Compare similarity: *UntestedXMLInput* (VA_{21}) is found both syntactically and semantically similar to *XMLDocument* (DA_1);

6) Select VA_{ik} (VA_{21} in our case) keys: The process continues to search VA_{ik} keys. For our example, the VA_{21} key list contains *UntestedDTDInput* and *UntestedXPATHInput* (related to the *has* relation);

7) Select DA: Domain experts study if the domain model involves any of the assets which are in the VA_{21} key list, but have not yet been identified as domain assets. For the example in Figure 6, the domain experts realize that the XMLDocument DA does involve a DTD, which in this case can be manipulated by the user (attacker), without any intermediary tests. Possible impacts include that XML parsers, which process the DTD, consume excessive resources, resulting in resource depletion. Therefore, the *UntestedDTDInput* VA is a VDA that is vulnerable and involved in the domain;

8) Add into DA list: The domain experts add *UntestedDTDInput* to the DA list;

9) Add into VDA list: The security experts add as well, *UntestedDTDInput* to the VDA list. The VDA list for this example is shown in Figure 8. The same reasoning applies to other keys, which we do not detail here for the reason of readability;

10) After adding the discovered DA and VDA into each list, i increments, now i is equal to 3, which is not lower than n (=3 initially). Therefore, the process stops.

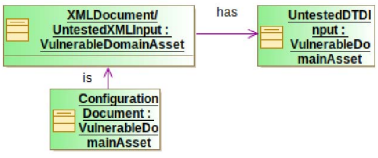


Fig. 8: An Excerpt of Vulnerable Domain Assets

As a result of this process illustration, we discovered the VDA *UntestedXMLInput* and *UntestedDTDInput*. *UntestedDTDInput* is not initially annotated as DA by domain experts. These two VDA are initially VA. Therefore, the threats that compromise these VA can be retrieved using the VA library presented in Section IV-A. As such, for *UntestedXMLInput*, the following 13 threats are identified: 1) XML Schema Poisoning (CAPEC-146), 2) XML Ping of the Death (CAPEC-147), 3) XML Entity Expansion (CAPEC-197), 4) XML Entity Linking (CAPEC-201), 5) Spoofing of UDDI/ebXML

Messages (CAPEC-218), 6) XML Routing Detour Attacks (CAPEC-219), 7) XML External Entities Blowup (CAPEC-221), 8) XML Attribute Blowup (CAPEC-229), 9) XML Nested Payloads (CAPEC-230), 10) XML Oversized Payloads (CAPEC-231), 11) XML Injection (CAPEC-250), 12) XML Quadratic Expansion (CAPEC-491) and 13) XML Flood (CAPEC-528). For *UntestedDTDInput*, there is only one threat identified: DTD Injection (CAPEC-228).

C. Discussion

1) Case study discussion: Whereas the Microsoft SDL threat modeling tool identified two threats for the case study, our asset identification process identified 14. Among these 14 threats, Spoofing of UDDI/ebXML Messages (CAPEC-218) and XML Routing Detour Attacks (CAPEC-219) belong to the same threat category of Spoofing, as the Spoofing of destination data store configuration file; Similarly, XML Flood (CAPEC-528), XML Ping of the Death (CAPEC-147), XML Nested Payloads (CAPEC-230), XML Entity Expansion (CAPEC-197), XML Quadratic Expansion (CAPEC-491), XML Oversized Payloads (CAPEC-231), XML Entity Linking (CAPEC-201), XML Attribute Blowup (CAPEC-229) and XML External Entities Blowup (CAPEC-221) belong to the same threat category Denial of Service, as the potential excessive resource consumption for web service or configuration file. As we can see, we identified more detailed threats comparing to the Microsoft SDL threat modeling tool. Moreover, our asset identification process discovered new XML Schema Poisoning (CAPEC-146) and XML Injection (CAPEC-250) threats, which are not found by the Microsoft tool.

A number of threats identified by our process come from proposing VA as new DA, for example *UnTestedDTDInput*. This enables identifying in-depth domain assets (that are also vulnerable), which otherwise may be overlooked.

As we can see, by integrating our asset identification process, we have found 14 threats, whereas sole Microsoft SDL threat modeling tool has found only 2 threat categories, which need further discussion and clarification during the brainstorming meeting. The average number of overlooked threats is very high as mentioned in [26], there is thus no guarantee that the brainstorming meeting can cover all 14 threats that we have found, because it depends highly on the security expertise, experiences and creativity of participants. The result of our asset identification process can thus be used as a checklist included in the brainstorming meeting to offer a guidance, to be complementary with Microsoft DFD and STRIDE based approach.

The DFD is data-centric, it focuses on the data flow between components of the same abstraction level. Each new template, which can be created with the tool, can represent a new abstraction level. However, DFD does not allow presenting **relations** among elements of **different abstraction levels**. That is why we model the case study with UML class diagram, because it allows modeling elements with different abstractions levels.

To integrate DFD into our process, the four DFD element types can be mapped to our asset reference model. As shown in Figures 5 and Figure 6, the process "web service" is mapped into the *WebService* domain asset, the data store "configuration file" is mapped into the domain asset *ConfigurationDocument*, and the "general data flow" is mapped into *depends* relation to show the interaction. The DFD diagram containing these three elements is mapped into the domain asset *Websphere7.0* together with *has* relations.

A limitation of this case study would be that we have not compare our asset identification process results with that of a real brainstorming meeting by industrial participants. This would be a future work to validate our approach.

2) General discussion: As we have seen in Section II, most of the existing threat modeling processes do not detail the asset identification phase. They usually consider it to be done through a discussion, usually of a non-structured, brainstorming type. The quality and quantity of the result of brainstorming meeting depends highly on participants. Moreover, such a discussion is highly creative and involves an important cognitive charge. By proposing a structured and detailed asset identification process together with the asset reference model, we help structure and guide this phase, which satisfies the Requirement 1 in Section II-C. This asset identification process can be reused in different domains, as the VA library contains VA that is domain-independent. It is worth noting that several activities of our asset identification process still need human expertise, such as "similarity comparison" and "search if a VA is involved in the domain", these two human tasks pose yet a much easier cognitive load than that of the entire brainstorming.

Other problems encountered in non-structured brainstorming sessions are that some details or system parts are overlooked, or the stakeholder input is not captured accurately. Hence, more in-depth threat modeling is typically performed afterwards by a security expert in isolation, which can be error-prone, as it is performed by manually iterating through a model, and with a lack of specific domain knowledge, such as a particular technology used in the system [16]. Our asset-based reference model can help consider both domain specific knowledge, by instantiating domain assets, and security knowledge, by extracting vulnerable assets. This two knowledge is shared by vulnerable domain assets (VDA), which can be established as a common vocabulary that can be understood by both experts, responding to the Requirement 3.

The concept of *asset* is easily understandable by non-security experts compared to the concept of *threat* together with that of *attack technique*. Identifying VA that can later derive threats thus helps bridging the gap during the collaboration between domain experts and security experts, satisfying Requirement 2.

V. Related work

In this section, we investigate existing works with a specific focus on the asset identification to deal with security issues, and compare them with our proposition in threat modeling.

Then, as the purpose of our proposition is to support collaboration between participants, we compare our study with other works focusing on collaboration in threat modeling.

A. Asset identification

Asset identification is an important step in numerous risk assessment (including threat modeling) methods, reviewed and compared by [13], [33], such as EBIOS, MEHARI, OCTAVE, IT-Grundschutz, MAGERIT, CRAMM, HTRA, NIST 800-30, RiskSafe Assessment and CORAS. For some of them, the concept of *asset* is defined very largely, rather vaguely, as anything that can have value to the organisation . Other methods try to separate the *asset* concept into several types, e.g. EBIOS into primary and supporting, or the ISSRM model into business and IT asset, the HERMENEUT approach [12] into tangible asset and intangible asset, etc. These separations help little, if any, the next phase of threat enumeration, while our approach does, because it considers the different perspectives between domain and security experts.

[4] proposes the notion of "secondary asset", the harm of which can cause harm to a "primary asset". This is captured by our reference model through the *depends* relation. In our case, the "supplier" is equivalent to the "secondary asset", and the "client" depending on the "supplier" is equivalent to the "primary asset". However, we can model extended chains of dependency relations between several assets, whereas the "secondary asset" does not allow this.

[19] proposes a security repository meta-model to store all the reusable elements. They add several concepts including "asset", based on the work of [30]. They indicate that the asset can be valuable or critical, but also vulnerable. However, they don't detail more about how to systematically distinguish each type of asset.

[24] identifies assets in the software architectural model, by mapping them from a system or organizational level. Their identification process is therefore focused on tracing assets from a development phase to another, whereas our identification process matches two different viewpoints.

B. Bridging the gap during the collaboration

[9] uses *anecdotes* and *scenarios* to express security knowledge and to reason about security in order to facilitate the communication among different stakeholders. *Anecdotes* are frequently used to communicate knowledge about real, concrete and specific security issues. However, *Scenarios* are even more widely used as a means of communicating security concepts, reasoning about security principles and justifying viewpoints. Weaknesses of *anecdotes* and *scenarios* are related to the difficulty in generalising their information content. That is to say that *anecdotes* and *scenarios* contain highly specific descriptions of particular events in a system, whereas security needs have to encompass the system as a whole. Yet, this approach only deals with requirements models, whereas ours may involve domain models from any and all phases of the development lifecycle. Moreover, the security details involved are fine-grained and difficult to generalise, whereas our reference model and process are aimed to be reused and deal with multiple levels of security abstraction.

[8] proposes a security ontology to resolve the communication problem. They took into account the entire infrastructure as asset which is physical and belongs to business domain. The ontology guarantees shared and accurate terminology in order to reduce misunderstandings. Comparing to their approach which aims at replacing the security expert, we introduce a collaborative process. As our abstract concept *asset* can be refined into domain asset (which can be understood by domain experts) and vulnerable asset (understood by security experts), the projection of general VA on DA, resulting VDA (understood by both), creates the possibility for better collaboration.

VI. Conclusion and Future Direction

Threat modeling is a result of a collaborative process involving many actors from different backgrounds. Despite its importance, the collaboration between domain and security experts to bridge the gap between domain modeling and threat enumeration phases is not trivial. One of the main reasons is that threat identification and enumeration is often a challenging task for non-security experts. Thus, domain experts have to rely on threat modeling processes, which may quickly turn into a complex task when these processes lack guidance and formalisation.

To address this limitation, we propose a reference model and a systematic asset identification process to facilitate the collaboration between actors. As a result, pertinent assets such as Vulnerable Assets are structured and Vulnerable Domain Assets are identified from security database CAPEC to improve the threat enumeration phase. Then, we have discussed how the proposed approach could be applied to structure the security knowledge base (CAPEC) and how the proposed process could be integrated with, and complementary to, the Microsoft SDL threat modeling process, using an appropriate case study. Results show the usefulness of our findings in identifying new assets and threats, and in bridging the gap between the domain and the security experts through the formalisation of the brainstorming activity.

The approach presented in this paper can be extended to become an aid system for the experts mentioned above, thus strengthening the bridge which facilitates their collaboration. To achieve such an aid system, the following objectives will have to be achieved, among others: 1) automating the security knowledge base extraction to offer appropriate guidelines to domain experts with modest security expertise, which is an ongoing work; 2) proposing a semi-automatic assistance based on the formalised reference model and the structured process that we are proposing, in order to suggest possible attack mitigation and/or security controls to the domain experts. This work is accepted and published in [23]. These two objectives guide our future work.

References

[1] ISO/IEC 21827:2008. Information technology – security techniques – systems security engineering – capability maturity model, 2008.

[2] Christopher J. Alberts, Audrey J. Dorofee, James Stevens, and Carol Woody. Introduction to the octave approach. 2003.

[3] R. Bayer and E. McCreight. Organization and maintenance of large ordered indices. New York, NY, USA, 1970. Association for Computing Machinery.

[4] K. Beckers, D. Hatebur, and M. Heisel. A problem-based threat analysis in compliance with common criteria. pages 111–120, 09 2013.

[5] Microsoft Corporation. Sdl threat modeling tool. security development lifecycle., July 2018.

[6] D. Dhillon. Developer-driven threat modeling: Lessons learned in the trenches. *IEEE Security Privacy*, 9(4):41–47, 2011.

[7] Marc Donner. Toward a security ontology. *IEEE Security and Privacy*, 1(3):6–7, May 2003.

[8] A. Ekelhart, S. Fenz, M. Klemen, and E. Weippl. Security ontology: Simulating threats to corporate assets. pages 249–259, 12 2006.

[9] I. Flechais and A. Sasse. Stakeholder involvement, motivation, responsibility, communication: How to design usable security in e-science. *Int. Journal of Human-Computer Studies*, 67:281–296, 04 2009.

[10] The OWASP Foundation. Owasp attack list, 2017.

[11] Martin Fowler. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Object Technology Series. Addison-Wesley, Boston, MA, 3 edition, 2003.

[12] E. Frumento and C. Dambra. The role of intangible assets in the modern cyber threat landscape: the hermeneut project. 5:2019, 02 2019.

[13] D. Gritzalis, G. Iseppi, A. Mylonas, and V. Stavrou. Exiting the risk assessment maze: A meta-survey. *ACM Comput. Surv.*, 51(1):11:1–11:30, January 2018.

[14] Michael Howard and Steve Lipner. *The Security Development Lifecycle*, volume 34. 06 2006.

[15] R Kamatchi and Kimaya Ambekar. Analyzing impacts of cloud computing threats in attack based classification models. 2016.

[16] Y. Koen, H. Thomas, V. Dimitri, S. Laurens, W. Kim, and J. Wouter. Threat modeling: from infancy to maturity. *New Ideas and Emerging Results, ICSE*, 2020.

[17] Loren Kohnfelder and Praerit Garg. The threats to our products. *Microsoft Interface, Microsoft Corporation*, 33, 1999.

[18] Gary McGraw. *Software Security: Building Security In*. Addison-Wesley Professional, 2006.

[19] D. Mellado, E. Fernández-Medina, and M. Piattini. A common criteria based security requirements engineering process for the development of secure information systems. 29(2), 2007.

[20] MITRE. Common attack pattern enumeration and classification., 2007.

[21] MITRE. Attck matrix for enterprise, 2015.

[22] N. M. Mohammed, M. Niazi, M. Alshayeb, and S. Mahmood. Exploring software security approaches in software development lifecycle: A systematic mapping study. *Comp. Stand. & Int.*, 50:107–115, 2017.

[23] MESSE Nan, CHIPRIANOV Vanea, BELLOIR Nicolas, EL-HACHEM Jamal, FLEURQUIN Régis, and SADOU Salah. An asset-based assistance for secure by design. *2020 27th Asia-Pacific Software Engineering Conference (APSEC)*, 2020.

[24] T. Rauter, A. Höller, J. Iber, and C. Kreiner. Asset-centric security risk assessment of software components. In *Workshop on MILS: Architecture and Assurance for Secure Systems*, 01 2016.

[25] Keunwoo Rhee, Dongho Won, Sang-Woon Jang, Sooyoung Chae, and Sangwoo Park. Threat modeling of a mobile device management system for secure smart work. *Electronic Commerce Research*, 13, 09 2013.

[26] R. Scandariato, K. Wuyts, and W. Joosen. A descriptive study of microsoft's threat modeling technique. *Requirements Engineering*, 20, 06 2013.

[27] N. Shevchenko, T. A Chick, P. O'riordan, Thomas P. Scanlon, and C. Woody. Threat modeling: a summary of available methods. *Software Engineering Institute. Carnegie Mellon University*, 2018.

[28] Adam Shostack. Experiences threat modeling at microsoft. 01 2008.

[29] Adam Shostack. *Threat Modeling: Designing for Security*. 2014.

[30] Guttorm Sindre and Andreas Opdahl. A reuse-based approach to determining security requirements. 05 2003.

[31] W. Stallings and L. Brown. *Computer Security: Principles and Practice*. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edition, 2014.

[32] J. Steven. Threat modeling - perhaps it's time. *IEEE Security Privacy*, 8(3):83–86, May 2010.

[33] A. Syalim, Y. Hori, and K. Sakurai. Comparison of risk analysis methods: Mehari, magerit, nist800-30 and microsoft's security management guide. In *Int. Conf. on Availability, Reliability and Security*, 2009.

[34] P. Torr. Demystifying the threat modeling process. *IEEE Security Privacy*, 3(5):66–70, Sep. 2005.

[35] K. Tuma, G. Calikli, and R. Scandariato. Threat analysis of software systems: A systematic literature review. *Journal of Systems and Software*, 144:275 – 294, 2018.

[36] Tony UcedaVelez. Real world threat modeling using the pasta methodology. *OWASP App Sec EU*, 2012.

[37] Anton V. Uzunov and Eduardo B. Fernandez. An extensible pattern-based library and taxonomy of security threats for distributed systems. *Computer Standards Interfaces*, 36(4):734 – 747, 2014.

[38] Wenjun Xiong and Lagerström Robert. Threat modeling – a systematic literature review. *Computers Security*, 84:53–69, 03 2019.

Using the architecture of Socio-Technical System to analyse its vulnerability

Paul Perrotin[1], Nicolas Belloir [2], Salah Sadou[3], David Hairion[5], Antoine Beugnard[4]
[1]Chair of Naval Cyber Defense, Ecole navale - CC 600, F29240 Brest Cedex 9
[2]Military Academy of St-Cyr, CREC St-Cyr, IRISA, Vannes, France
[3]University of South Brittany,IRISA, Vannes, France
[4]Institut Mines-Télécom Atlantique, Lab-STICC CNRS UMR 6285, Brest, France
[5]Naval Group, France
contact : paul.perrotin@ecole-navale.fr

2022 17th Annual System of Systems Engineering Conference (SOSE) | 978-1-6654-9623-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/SOSE55472.2022.9812648

***Abstract*—Today, most complex and large systems, such as healthcare systems, integrate the physical and human aspects of computing and networking. They constitute a system of systems with a socio-technical part. The human element is nowadays one of the most important attack vectors in the context of systems of systems (SoS). In this context, the estimation of the impact of human vulnerability on these systems enables for a more secure design and for an integration of the system development cycle with security concerns. This approach is known as "security by design". To improve the resilience of these SoS with respect to the vulnerability that humans bring, it is necessary to be able to estimate the impact that an individual can have on the system. In this article, we present an approach that enables to assess the impact of human vulnerability on a SoS composed of humans, which in this case will be a social technical system (STS). We propose to use behavioral models to model the propagation of a human vulnerability in a STS. We propose to use different models in order to capture the plurality of attacks on STS.**

***Index Terms*—Human vulnerability, human models, cybersecurity, vulnerability propagation**

I. Introduction

In an interconnected world, it is commonly admitted that combining systems should not just offer the sum of their functionalities, but also exhibit new (and sometimes undesired) behaviors and performances [4]. This is particularly emphasized when parts of the system have their own intelligence and creativity, which is the case for humans. Such systems are often referred to as "Socio-Technical Systems" (STS) [8]. Thus, these systems are software intensive and they interact with a set of actors such as humans and organizations. So, we can call them Systems of Soscio-Technical Systems (SoSTS).

In the past, several studies have been dedicated to the rigorous design and verification of these systems [1]. But what about cybersecurity challenges raised by SoSTS? The complexity of these challenges lies in type diversity of the raised vulnerabilities: i) those related to software-intensive systems; ii) those raised by the emerging behaviors, which are the result of the collaboration between the software-intensive systems; iii) human vulnerabilities; iv) and, those raised by the combination of the precedent types of vulnerabilities. It seems obvious to study the vulnerability of systems of systems (SoS) made up of the same type of systems (software intensive or human) first before combining the two types. A lot of work has focused on decreasing system vulnerabilities [5] where the constituents are software-intensive systems. In this paper, we deal with the vulnerability of SoS where the constituent are humans. After this step, it will be necessary to tackle the problems linked to the combination of the two types of vulnerabilities. This is out of the scoop of this paper.

In a recent work, we explored works on human characteristics coming from the field of human sciences. So, we proposed a model to estimate the vulnerability of a human operator in an organization (SoS made up of humans) [15]. During our experiments with our industrial partner[1], we found that the vulnerability of an operator can make another operator vulnerable from the same SoS. Sometimes the impacted operator has an important and sensitive role, whereas the initially vulnerable operator did not have a role considered important in the SoS. Thus, it is important to study this phenomenon of propagation of a vulnerability through the architecture of an SoS. Actually, it is not the vulnerability of an operator that is propagated, but rather her/his vulnerable state which, in a way, contaminates the state of the other operators.

Thus, in this paper we propose an approach to estimate the propagation of a human operator's vulnerability through the architecture of an SoS of which she/he is a constituent. The architecture is used to explicit the relationships between human operators which are propagation prone. In this study we will focus on three important types of relationships: transmission of documents, delegation of objectives and human specific relationships. These are the ones that are the most studied in the humanities literature and that allow us a sufficient level of formalization for our study. The interest of our work through the study of the propagation of human vulnerabilities in an SoS is to allow the assessment of its level of resilience with respect to these types of vulnerabilities. In the following section, we briefly describe our security-oriented architecture description language, named HoS-ML, which allows to model an SoS with a security orientation. In Section III we describe our approach for estimating vulnerability propagation in an SoS

[1]Naval Group: European Leader in naval defense.

where all constituents are humans. In Section IV, we evaluate our approach through a case study from the industrial world. Before concluding this paper in Section VI, we will discuss some related work in Section V.

II. Describing an STS

Knowing the vulnerability of a human operator to a cyber-attack makes it possible to evaluate its impact on the system and thus the resistance of the system itself. In previous work [15], we defined a specific approach to do so. We present here the language we have developed to model an STS by including human vulnerability in the representation of the operators.

A. HoS-ML presentation

We first proposed a human model to describe a human operator in an STS. This model is based on human properties identified in the literature. These properties are organized into two types: direct factors and indirect factors. The direct factors allow characterizing directly a human operator and the indirect factors specify the context in which this operator operates. Examples of direct factors are skill, reliability or emotional stability. Examples of indirect factors are management, task exigencies, or communication. We invite the reader to consult [15] for a complete description of direct and indirect factors.

In a second step, we specified an architecture description language dedicated to architects, called HoS-ML, allowing describing the architecture of an STS. The models realized through this language describe the human operators, their direct and indirect factors, and the relationships between them. First, architects specify human operators as they want them to be. Such a human operator is called a "role" (Nurse role in Figure 1 for instance). It represents the desired profile that the human operator who will occupy a given position should have. Next, the architect describes the relationships between the roles. Currently, three types of relationships are possible: document transmission (represents an exchange of data between two roles - in green in Figure 1), goal delegation (represents the fact that one role shares a common goal with another - in red in Figure 1) and specific human relationships (represents relationships such as collaboration or supervision in a task for example).

In Figure 1, two roles are specified : a nurse and a Lab Technician. Nurse role needs to be played by a human operator with high-level skills (5/5), an emotional stability at "stable" and a confidence level of 4. These properties are direct factors. For this role, indirect factor are a CB management and both a resource level of 4/5 and position level of 4/5. Lab Technician must provide to nurse results of blood analysis. This data transmission must be confidential and must preserve data integrity. Nurse delegates the objective "Save the patient life" to Lab Technician. For the latter, the objective becomes "analysis blood".

In a third step, architects integrate "actors" in the model. Actors is an implementation of a specific role with its own values for direct and indirect factors. For instance, as shown by Figure 1, the role "Nurse" was specified with its factors, for instance the value *a* for the *X* factor. Architects add Alice. The latter is the actor implementing the Nurse role. Architects can vary factors for her. For instance, she may have the value *b* for the *X* factor. The simulation will evaluate this difference between the two values to calculate if it makes a new vulnerability. Thus, architects can evaluate situations such as what happens if the human operator in a specific position is to be characterized by a specific factor and ultimately has a different value for this factor. For example, in Figure reffig:HoS-ML-Example, the direct factors for the actor Alice are different from those specified for the Nurse role.

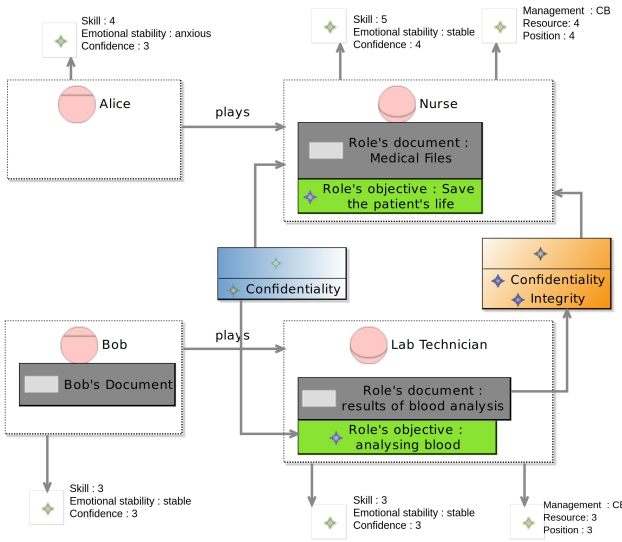


Fig. 1. An example of a STS architecture specified using HoS-ML

B. Vulnerability detection

The method we propose to estimate human operator vulnerability is based on the following points:

1) The difference between the operator's direct factors and those needed by the corresponding role with respect to the indirect factors. The difference between the operator and the role for the same direct factors can increase the probability that the former is a source of vulnerability in the system. For example, if an operator has less skill than expected
2) A link between different factors can increase or decrease an existing vulnerability. For example, some management approaches can help a person to stress down.

To implement the previous points, we have decided to use Bayesian network (BN) [14]. This choice was made because of the advantage that BN can give a distribution of the probabilities on a model having many links and nodes. It can also learn from existing data that can be more accurate than using probabilities extracted only from the literature and corresponding to a particular case. The construction of the BN can be seen in Figure 2.

As already said, the difference between the operator's factors and those expected by the role allows us to detect a possible vulnerability. In order to allow a good adjustment to reality, we need to allow the user to enrich the BN with existing data from the application domain Once the BN has been resolved, it gives a probability according to the level of vulnerability. Indeed, we have chosen to rate the vulnerability from 1 to 5 to have a certain consistency between the threat and the impact. The BN gives us the probability for each level of vulnerability and the one with the highest probability is given to the architect.

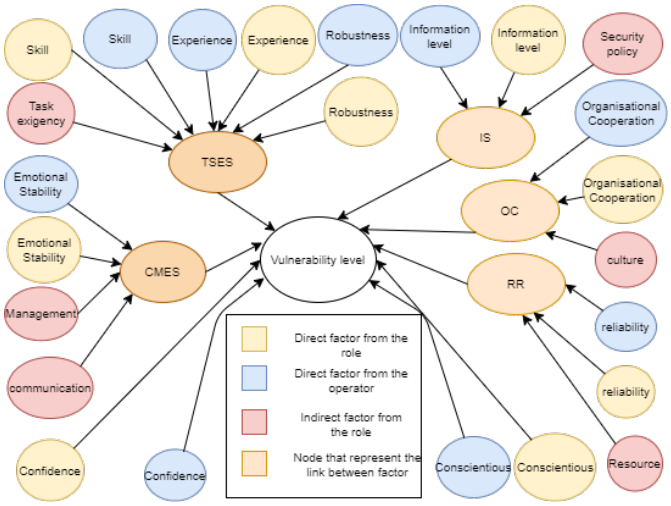


Fig. 2. Bayesian network for estimating operator's vulnerability Level.

III. Human Vulnerability Propagation

To calculate the human vulnerability in a STS, we first chose an ad hoc approach. This first approach, which was made in consultation with industrial experts, allowed us to define a propagation table I based on the level of vulnerability of an individual as well as his impact. We have 3 types of possible links for contamination. The choice of these links is based on our HOS-ML language : we use the structural and relational links of the operators as a means of contamination.

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) $\neq$ *Good*	impact(A) $>$ 2 & vulnerability(A) $>$ 2 & CMES(B) $\neq$ *Good*

TABLE I
Neighbour contamination conditions according to link type.

The impact is an important criterion, because it is not enough to be vulnerable to endanger the STS. It is also necessary that the operator is able to impact his system either by his functional links or by his hierarchical position. We calculate the impact as follows: $Impact(A) = (PositionOf(A) + ConfidenceIn(A))/2$

To calculate the impact of our operator on our STS we consider in an equivalent way the hierarchical position and the trust invested in this role by the architect. Indeed, for an operator to have an important impact on the STS, it is necessary that he has an important hierarchical position, but also that the system has a need for trust in the operator. This trust was expressed through the manipulation of sensitive information or tasks essential to the system.

Our approach will have three types of links for propagation in a system. The first two links are the information and goal delegation links which are links that are part of our language. The third link is the link that represents the emotional contagion linked to stress. Indeed, the emotional contamination by stress is a possibility when there is a personal link between two people [2]. In this first approach, we have chosen to allow the propagation in all types of links: structural links being part of the system representation, individual links being able to belong to each of the operators present in the system. The feedback obtained from the experimentation and from experts [15] showed that this approach was generic and allowed an approximate evaluation of the propagation. We have therefore sought to improve it by allowing for adaptability of the propagation according to the type of scenario. The propagation of a cyber attack in a STS can indeed differ depending on the attack. To test this first propagation diversification, we first focused our propagation model on two types of attacks:

- Attacks on false information, the propagation here will aim to represent how this false news is spread in the system
- Attacks on emotional contamination, here the propagation will be used to represent the specific impact of stress on the system in a case of crisis management. The first propagation model is therefore the one that focuses on false information in a system.

The first model that we have chosen to present a propagation of fake news is present in [7]. This model will allow to represent in a generic way the propagation of fake news in a group of individuals. From [7] we have set up the following table II representing the propagation according to the impact of the individual in our system. Here only the link of the information transmission can be a vector of this attack. This specificity is due to the fact that the link of information transmission is the link representing the possibility between two individuals to exchange and thus to contaminate each other if ever information is bad. To obtain the probabilities present in the table II , we apply a conditionality to who was reached and who was not by the propagation. This gives us for the first line that the possibly vulnerable operator (here weakly) is not considered to be the majority vector of propagation. For the second line, it becomes the majority vector so we apply here the conditional probabilities to obtain the values of the line. For the third line, we do the same by shifting the propagation from the individual to the next.

Impact & vulnerability	*Document transmission first link*	*Document transmission second*
impact(A) $>$ 2 & vulnerability(A) $>$ 2	15%	4%
impact(A) $>$ 3 & vulnerability(A) $>$ 3	65%	33%
impact(A) $>$ 4 & vulnerability(A) $>$ 4	99 %	50%

TABLE II
NEIGHBOUR CONTAMINATION CONDITIONS ACCORDING TO LINK TYPE.

The second propagation model that has been put in place is that of emotional contamination. Indeed, capitalizing on the panic of operators who are part of a system can be a good way for an attacker. The propagation model implemented to represent this emotional contamination in a system is extracted from [12]. The table III and the application of the data coming from [12] apply our language. We have here two types of links that can propagate a vulnerability of this type. The first link is indeed the link between individuals, the second dog and him the delegation by objective which is going to represent the possibility of an emotional contamination between co-workers during the arrest of a task. To first check the relevance of these two models, we confronted the expert opinions obtained during the first experimentation. These new propagation models are close to the results obtained with the ad hoc model. This proximity shows that it can be used in addition to the ad hoc model if we wish to model a specific propagation for a type of attack. Moreover, it is possible to add other propagation models that will be used during the stimulation of the system against an attack to evaluate more precisely the resistance of the system to a type of attack.

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 III
NEIGHBOUR CONTAMINATION CONDITIONS ACCORDING TO LINK TYPE.

For the probabilities that we have held in the table III. Specifically, we have taken the model related to high stress conditions. To transpose this model to our vulnerability model, we have transposed our vulnerability to social pressure. The social pressure that as a vulnerability of a value on a scale of 5. For the contamination of the second link, we consider the vulnerability as a lower level. The equation we used is the following:

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

IV. CASE STUDY

To evaluate the proposed propagation model, our industrial partner provided us with a groundthrue case study. The aim is to compare the contribution of the new models against what we have already evaluated with the support of the same industrial partner. This case study is an SoS representing the fight against maritime piracy.

The studied STS is composed of 5 roles having interconnections between them and 5 actors playing the roles and representing human operators. The representation of the architecture using our language is visible in the figure 3.In this figure we can see the different roles (officer, intervention leader...) in the center. The roles that can be seen on the right side of the figure are represented here by the actors named from Alice to Eve. The elements in green represent the different objectives that each role has. The elements in gray represent the information that the roles possess. The blue boxes represent the delegations of objectives either when several roles collaborate around the same objective this box is used as a link between the roles. The transmission of documents between roles is represented here through the orange case and the links which emanate from it. During the simulation, we both varied (i) the values of the human factors of the actors (regarding the values defined by the roles), and (ii) applied attacks of different levels of complexity on these actors. Thus, we have both measurements on the probabilities of vulnerability induced by the gaps between roles and actors, and also the impacts when a threat targets one of these actors. We limited the study to reasonable gaps (not considering extreme variations). Among these gaps, two operators were identified to be vulnerable: the first one was by the difference between the values of the human factors. The second one by the propagation of the vulnerability generated by the first one to the second one. The role of the first one was specified as having a low-impact level on the system. Nevertheless, despite this, a vulnerability was identified as to be contagious and to be able to contaminate another actor having a very higher potential impact to the system. This implies that an attack against the first actor can cause significant damage to the system even if its responsibility level is low. The results we obtained with the initially ad hoc propagation seemed satisfactory with regard to the feedback from the experts. But they were not precise enough to allow differentiating the type of propagation according to the operational scenario carried out. The available representation is shown in the Figure 3.

To represent the two different types of propagation visible in the figure 3, we have generated two different scenarios, each with a specific type of attack. In the first scenario, we have tried to simulate an attack linked to false information, in this case with no AIS trace. The second scenario represents a case of strong emotional transmission, which in our case corresponds to an attack linked to blackmail on an individual generating stress which in the long term can contaminate other people. These propagation can be seen in the Figure, in red the first scenario on blackmail and in cyan the second scenario on AIS traces. We can thus observe with the objectives and documents present for each of the roles that the impact can be important with regard to the first attack. But this is weaker for the second scenario, because the role of the monitor Chief has only one transmission of documents towards the intervention Chief. It can therefore potentially only misinform this role.

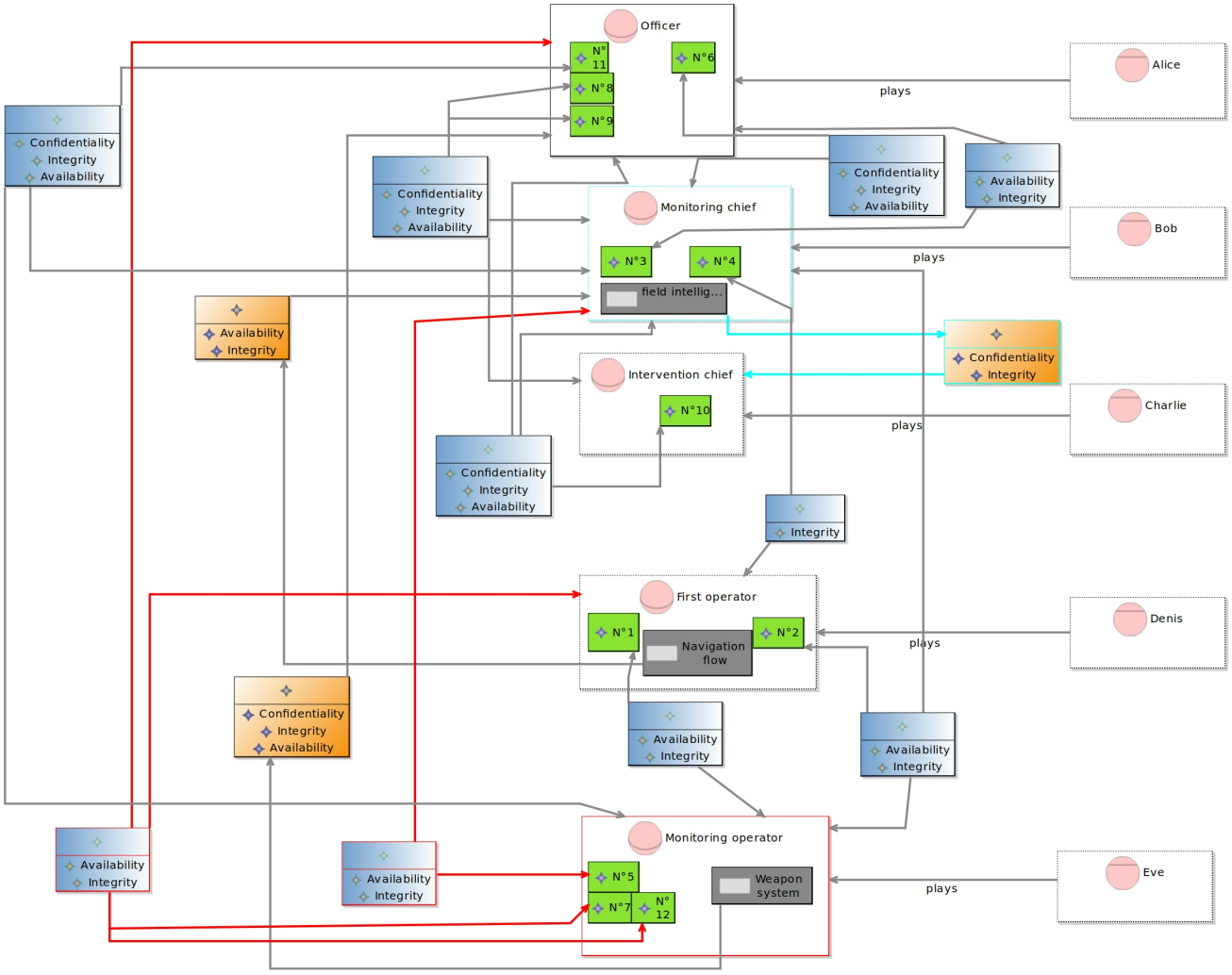


Fig. 3. The maritime case study representation with HOS-ML.

The propagation obtained after simulation remain in agreement with the judgment of our industrial experts on the cases previously obtained. However, they bring a specificity according to the type of threat that will be set up in the scenario and thus better allows evaluating the real impact of an individual in a system.

V. Related work

Our work is mainly related to works on human modeling and those on vulnerability propagation. Thus, in the following we discuss these two kinds of research works. Assessing the vulnerability of a human as an actor in an STS (human operator) requires taking her/his traits into account and considering each one as a vulnerability factor. For instance, a human operator who is resilient to stress will be able to work in stress-generating positions. Unlike traditional security analysis approaches, which rely on technical aspects, for human ones we need to rely on psychological and sociological aspects. The latter is often related to the position of the human operator in the STS.

In the literature, few studies have focused on this problem. We have identified two types of approaches: organizational approach [11] and more systemic one [5], [13]. In [11], the author established a model to describe humans in the management of risk-investment constructs, in security investments and in constructive feedback situations for security incidents. The approach presented by the author results in some feedback on incidents related to human factors. This feedback helps managers to have the best managerial policy, according to the targeted person. They describe what they have called ”human factors” in a model with an organization-level management in order to be more resistant to social engineering attacks. The human model is organized into two parts: i) direct factors, describing the characteristics of a person. ii) indirect factors, describing the constraints of the human role in her/his environment. This approach describes the human operator with properties. It makes it possible to specify *a posteriori* which property is at the origin of a given incident. This approach is very interesting because the proposed model

aims to characterize what is expected by a person at a given position. We used this work and adapt it to the case of human vulnerability detection in STS. Indeed, we expanded the list of proposed properties refined others.

The propagation of human vulnerability is not a widespread concept in the field of cyber security. Therefore, we had to rely on related work in cyber security and adapt them while taking into account elements from work from the human sciences. Indeed, in cyber security there is the notion of propagation of an attack through probabilistic models such as models using Bayesian networks [6] or models using machine learning [3] to deduce possible attack paths in a complex system. The missing element of these approaches is that they are based only on technical attacks and do not address human factors. Moreover, these models are based on well quantified elements that are cyber attacks and of which the literature gives several elements for learning models [9].

In the fields that can approach the notion of propagation of human vulnerability, there is a part of the literature on human sciences devoted to social imitations, in particular those related to the prediction of these imitations and the study of their functioning [10]. In the models that inspired us in the first instance, there are in particular those that allow the prediction of violent behavior, particularly crimes [16]. In this case there is a similarity with the notion of propagation of a behavior that can harm a system. We then looked for different models that could approach the themes related to the field of cyber security. Other works that can be considered as related to the propagation concept concerns those related to emotional contamination. In particular, we considered to those tackling problems of contamination linked to the stress and social behaviour [2]. Emotional contamination can, of course, be useful when cyber attacks generate a case of panic or crisis management that increases the stress of the various operators. We chose to combine these two approaches by taking the notion of propagation such as a cyber attack and using the human sciences models to do so.

VI. Conclusion

In this paper we have outlined the problem of contaminating the spread of human vulnerability in a system of systems (SoS) where humans are also constituent elements. To focus only on human vulnerabilities, we considered SoS with only human constituents. We have identified several propagation models that can be applied to human vulnerability. Then, we have integrated them into our HOS-ML language to allow a better consideration of the propagation of human vulnerability to specific threats.

The integration of this model, in our case study, has allowed us to provide a better evaluation according to the scenario that has been exploited. Thus, it showed that the proposed approach allows a better evaluation of the impact of a vulnerability on technical social systems. Moreover, as these models are currently limited to two types of propagation, they can be enriched or changed depending on the propagation model chosen by the architect when defining the architecture.

In a future work, we plan to integrate the technical part of the systems into our security analysis oriented architectural description language. This will allow us to combine human and systemic vulnerabilities and simulate the spread of a realistic cyber threat in systems of socio-technical systems.

Acknowledgments

This work is funded by the Chair of Naval Cyber Defense[2]: Ecole navale, ENSTA Bretagne, IMT Atlantique, Thales and Naval Group.

References

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

[2] Niall Bolger, Anita DeLongis, Ronald Kessler, and E. Wetherington. The contagion of stress across multiple roles. *J Marriage Fam*, 1989.

[3] A. L. Buczak and E. Guven. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys and Tutorials*, 2016.

[4] Paul Schreinemakers Sanford Friedenthal Sky Matthews David Nichols Christopher Oster Taylor Riethle Garry Roedler Emma Sparks Heinz Stoewer Christopher Davey, Paul Nielsen. Systems engineering vision 2035 - engineering solutions for a better world, 2021.

[5] Fabiano Dalpiaz, Elda Paja, and Paolo Giorgini. *Security Requirements Engineering: Designing Secure Socio-Technical Systems*. MIT Press, 2016.

[6] Nan Feng, Harry Wang, and Minqiang Li. A security risk analysis model for information systems: Causal relationships of risk factors and vulnerability propagation analysis. *Information Sciences*, 2014.

[7] Sharad Goel, Ashton Anderson, Jake Hofman, and Duncan J. Watts. The structural virality of online diffusion. *Management Science*, 2016.

[8] A. Gregoriades and A. G. Sutcliffe. Automated assistance for human factors analysis in complex systems. *Ergonomics*, 2006.

[9] Hanan Hindy, David Brosset, Ethan Bayne, Amar Kumar Seeam, Christos Tachtatzis, Robert Atkinson, and Xavier Bellekens. A taxonomy of network threats and the effect of current datasets on intrusion detection systems. *IEEE Access*, 2020.

[10] Stanley Milgram, Leonard Bickman, and Lawrence Berkowitz. Note on the drawing power of crowds of different size. *Journal of Personality and Social Psychology*, 1969.

[11] Reza Mortazavi-Alavi. *A Risk-Driven Investment Model for Analysing Human Factors in Information Security*. PhD thesis, University of East London, 2016.

[12] Mehdi Moussaïd, Mubbasir Kapadia, Tyler Thrash, Robert W. Sumner, Markus Gross, Dirk Helbing, and Christoph Hölscher. Crowd behaviour during high-stress evacuations in an immersive virtual environment. *Journal of The Royal Society Interface*, 2016.

[13] Elda Paja, Fabiano Dalpiaz, and Paolo Giorgini. Modelling and reasoning about security requirements in socio-technical systems. *Data Knowl. Eng.*, 2015.

[14] J. Pearl. Bayesian networks: A model of self-activated memory for evidential reasoning. In *Proc. of Cognitive Science Society*, 1985.

[15] Paul Perrotin, Nicolas Belloir, Salah Sadou, David Hairion, and Antoine Beugnard. Hos-ml: Socio-technical system adl dedicated to human vulnerability identification. In *26th International Conference on Engineering of Complex Computer Systems (ICECCS 2022)*. IEEE Computer Society, 2022.

[16] Martin B Short, Maria R D'orsogna, Virginia B Pasour, George E Tita, Paul J Brantingham, Andrea L Bertozzi, and Lincoln B Chayes. A statistical model of criminal behavior. *Mathematical Models and Methods in Applied Sciences*, 2008.

[2]Chaire de Cyberdéfense des Systèmes Navals: www.chaire-cyber-navale.fr

Characterizing Fake News: A Conceptual Modeling-based Approach

Nicolas Belloir[1,2][0000−0002−0163−8757], Wassila Ouerdane[3][0000−0002−4515−7461], and Oscar Pastor[4][0000−0002−1320−8471]

[1] CREC St-Cyr, Académie Militaire de St-Cyr Coëtquidan, Guer, France
[2] IRISA, Vannes, France
[3] MICS, CentraleSupélec, Université Paris-Saclay, Gif sur Yvette, France
[4] PROS Research Group, Universitat Politècnica de València, Spain
contact : nicolas.belloir@irisa.fr

Abstract. For some time, and even more so now, Fake News has increasingly occupied the media and social space. How identify Fake News and conspiracy theories have become an extremely attractive research area. However, the lack of a solid and well-founded conceptual characterization of what exactly Fake News is and what are its main characteristics, makes it difficult to manage their understanding, identification, and detection. This research work advocates that conceptual modeling must plays a crucial role in characterizing Fake News content accurately. Only by delimiting what Fake News is will it be possible to understand and manage their different perspectives and dimensions, with the ultimate goal of developing a reliable framework for online Fake News detection, as much automated as possible. To contribute in that direction from a pure and practical conceptual modeling perspective, this paper proposes a precise conceptual model of Fake Newss, an essential element for any explainable Artificial Intelligence (XAI)-based approach that must be based on the shared understanding of the domain that only such an accurate conceptualization dimension can facilitate.

Keywords: Conceptual Modeling · Characterization · Fake News · Explainable Artificial Intelligence

1 Introduction

Conceptual modeling is a discipline that contributes rich and diverse results when applied to well-understand the conceptual support of a particular domain of interest. One domain especially interesting nowadays is the Fake News one. Although Fake News is not a new phenomenon [20], questions such as why it has emerged as a global topic of interest and why it is attracting increasingly more public attention are particularly relevant in our times. The leading cause is that fake news can be created and published online faster and cheaper when compared to traditional news media such as newspapers and television [18]. In addition, recent discussions of higher education's failure to teach students how to identify Fake News have appeared in leading newspapers [14].

Conceptual Modeling should play an essential role to understand and communicate what Fake News is. In a sound Information Conceptual Modeling context, a correct data management of Fake News must be supported by a precise conceptual characterization of "what" a Fake News is. This is where Conceptual Modeling becomes a crucial actor. If an information structure must represent a conceptualization, the entities that represent that conceptualization must be explicitly determined. Any information system intended to register information about Fake News must identify in detail what are the relevant entities that conceptually characterize the different dimensions that must be considered to treat Fake News data correctly. Ontologically speaking, a precise ontological commitment that involves a precise identification of the relevant entities that constitute the conceptualization must be stated. Our previous work [5] has focused on exploring what are the concepts that should be considered for achieving that purpose. In this paper we introduce the result of such an ontological analysis, by presenting a conceptual model of Fake News. This is the main contribution here addressed, by facing a fundamental question regarding the terminology and the ontology of Fake News: what constitutes and qualifies as Fake News?

To achieve that goal, in Section 2 we discuss the literature and show that the views on the concept of Fake News are not unified. In order to propose the conceptual model of FN, the Section 3 summarizes the key notions that need to be considered when the goal is to provide a robust conceptual characterization of the notion of FN. These key notions are derived from a previous work. As a consequence, a precise definition of Fake News is proposed, and a Fake News Conceptual Model is presented in Section 4. A discussion of the application of this conceptual model as an initial building block for an XAI approach is provided in Section 5. Concluding remarks and the list of used references complete the work.

2 Related Work

In recent years, different works were interested in studying and understanding the nature of information encompassed in Fake News. Indeed, to help online users identify valuable information, there has been extensive research on establishing practical and automatic frameworks for online Fake News detection [2, 25, 24].

An important element is to be able to identify very clearly what is a Fake News or what are the principal features characterizing it. However, what we can notice first in the literature is that the concept of Fake News is still ambiguous, and the frontier between the definition of Fake News and other related concepts, such as mis-information, des-information, hoax news, propaganda news, etc., is blurred. Indeed, as it is illustrated by some categorization examples [20, 21, 11], it is not always clear or precise how these different concepts are related, or how we can distinguish between theme. On the other hand, as we can see in Table 1, several definitions of Fake News have been proposed in the literature, most of which include falsehood and the news form as common factors. Even if it can be argued that there is some common intuition on what a Fake News is, what we

can note is that it is difficult to have a consensus and a unified vision on what "exactly" -conceptually-speaking- a Fake News is.

Definitions	Reference
Fake News is a news article that is intentionally and verifiably false.	[18]
Fake News are intentionally false news published by a news outlet	[25]
Fabricated news articles that could be potentially or intentionally misleading for the readers	[15]
News articles that are intentionally and verifiably false, and could mislead readers	[1]
Fabricated information that mimics news media content in form but not in organisational process or intent	[12]
Fake News are fabricated stories presented as if they were originating from legitimate sources with an intention to deceive	[10]

Table 1. A Sample of Fake News Definitions

A recent work has proposed a first step towards a characterization of Fake News [13]. It introduces a taxonomy of operational indicators in four domain (message, source, structure, and network) to distinguish seven types of online content under the label of "Fake News" (false news, polarized content, satire, etc.). The proposed characterization is of interest, but it is not based on a precise conceptual model, which is our contribution in this paper. As a representation that captures the conceptualization of a person's understanding of a domain, a conceptual model is the natural strategy for obtaining a reliable representation of the domain that is used by human users to support communication, discussion, negotiation, etc. In our context, it allows us to define a domain with specific and precise semantics. Moreover, the conceptual model will expose the relationships between the concepts composing the Fake News in a more informative and robust way, which will offer a reliable and practical means for the detection or even automatic generation of Fake News. In this line, [22] have proposed a conceptual model to examine the phenomenon of Fake News. Their model focuses on the relationship between the creator and the consumer of the information, and proposes a mechanism to determine the likelihood that users will share their Fake News with others. In contrast, in this paper, we are particularly interested in the conceptualization of the content of Fake News.

A further advantage of relying upon a conceptual model is its ability to facilitate building well-justified and explainable models for Fake News detection and generation, which, to date, have rarely been available. Works as [4] explores and integrates the use of ontologies (OWL-based) trying to detect fake news on social media by identifying contextual features for news articles. However, as it was emphasized by [25, 23], despite the surge of works around the concept of Fake News, how one can automatically assess news authenticity in an *effective* and *explainable* manner is still an open issue, especially due to the lack of the precise conceptual characterization of the Fake News concept that this paper advocates.

Facilitating explainability and interpretability has been of great interest in artificial intelligence and machine learning research (see for instance [6, 7]). Indeed, eXplainable Artificial Intelligence (XAI) is recognized as a major need for

future applications. It aims to produce intelligent systems that enhance the confidence of users to understand the underlying reasoning and automations [9, 3]. For Fake News context, to our knowledge, works involving explainability feature within Fake News detection methods are still at their beginning [17].

In this paper, we propose an original approach to contribute to understanding what is a Fake News. Our proposal is novel at different levels. First, we offer to characterize Fake News content by relying on a Conceptual Model. A conceptual understanding of Fake News will help us distinguish them better and rule them from real news. A well-grounded conceptual characterization would make feasible to go beyond what the classical approaches normally do, by opening the door to design a Fake News generation process fully guided by the Conceptual Model. More precisely, we follow the XAI-based process proposed by [19] to facilitate building well-justified and explainable models for Fake News generation. Our aim is to propose an approach that is understandable, trustable and manageable to humans, as suggested in [8]. More specifically, the different steps suggested by [19] are: (i) Get a shared understanding of the domain, (ii) Understand the task and select the right scope, (iii) Collect the right data and improve its quality, (iv) Select AI techniques that deliver results, (v) Generate good explanations, and (vi) Evolve the solution over time. Our contribution establishes the foundation of such a process by solving the first, essential step of getting a shared understanding of what a Fake News is by introducing a precise conceptual model of Fake News. From that sound conceptual basis, the rest of the proposed XAI process can be applied in a reliable way. The explainability with our approach is conceptually guided by the conceptual model which conforms the core of the contribution: to have an ontologically well-grounded definition of what a Fake News is, which is directly derived from the conceptual model.

3 Characterization of Fake News

We briefly present our definition of the concept of Fake News and a summary of the essential semantic elements to be considered in the characterization of the concept of fake news. Both have been proposed and discussed in a previous publication [5] and we invite the reader who wants more details to consult it.

Definition 1 (Fake News). *A Fake News is false but verifiable news composed of false facts based on real ones. Drafted in a way to trigger an emotional load, it aims to deceive its readers and influence their opinion through an implicit conclusion.*

Among the highlights that have been identified, one concerns the origin of fake news. Indeed, in the same way that a cyber-attack is rarely the work of a single person but rather that of an organized group or even a state service, Fake News is rarely created by an isolated person. However, identifying the chain of creation of a Fake News allows us to understand what objective it serves, and therefore to better understand and counter it. Thus, a conceptual model of Fake News must be able to take into account the entities that were involved in the

creation of a Fake News and the different levels of decision-making involved. For example, at the moment Fake News target pro-Ukrainian political figures. This is part of an operational context aimed at manipulating pro-Ukrainian opinion and part of a strategic vision that seeks to justify the Ukrainian-Russian war.

Fake News is usually built on a distortion of reality. Representing this distortion is a second point of interest for the characterization of a fake news. Indeed, even if we can find fake news totally false, most of the time they are built by mixing true facts and false facts. This increases its credibility. The thinner the line between the two, the more real the fake news seems. Here, by fact, we mean the facts reported by the Fake News, and not facts that would be proven facts, that is to say facts that really happened. We call the latter real facts. The real facts can take different forms. For instance, real facts can come from a political statement, an event that occurred, or real data such as a photo or video. In the same way, different types of false facts exist. Here, what we call false facts are the elements of disinformation propagated by the fake news. Indeed, as mentioned earlier, false facts may not be based on any real element. This makes it less credible. Other times, the false fact may be constituted by distortion of real elements. Photos or videos can be altered, articles can be falsified. Finally, one can take them out of their context of real facts for example, or associate several real facts but independent of each other.

In order to reinforce the credibility of a Fake News, their authors often use an "authority". The latter can be of different nature but its objective is always to reinforce the realistic side of the Fake News by relying on a reassuring element. This authority can be of three types: internal, external or false. In the first two cases, the authority is real. For the first one, the authority can be a person or an entity in direct link with the subject carried by the Fake News. In the second case, it may be a reference to a historical authority whose word or deed is considered true and safe. Finally, in the last case, the authority may be totally invented or its field of competence may have nothing to do with the information carried by the Fake News.

Fake News is constructed to influence the opinion of its target by generating an emotional charge. The reaction of the target leads him to draw certain conclusions and to change his opinion on a subject. That is the objective. Identifying the latter and knowing who the Fake News is aimed at is therefore a challenge in order to try to counter it. Thus, in a conceptual model, three aspects need to be addressed: the target, its opinion on a topic and the way in which the Fake News will change it. It can seek to strengthen or weaken it. We talk about the goal of Fake News. The human mind can be seen as a two-tiered system. The first is instinctive, almost automatic. It reacts to perception, to emotion. The second is slower, more analytical. Fake News takes advantage of these two systems. It generates an emotion which makes the first system react. Then, in a second step, the second system analyzes the reaction to the emotion and to what generated it and draws conclusions. A Fake News must therefore be able to trigger an emotion. It does this by a catalyst mechanism. Then, the target of the Fake News reacts to this emotion and goes where the Fake News wants to

take him. This is what we call an emotional load. Without it, the brain analyzes the fake news in an analytical way and not in an instinctive way. The emotional charge can therefore be seen as the necessary precondition for fake news to work. A conceptual model must therefore be able to capture this mechanism.

The conceptual model that we present is articulated using the three main dimensions of: (i) identification of the origin of the Fake News (the 'attacker' dimension), (ii) the relationship between true and false fact (the 'fact' dimension), and (iii) the target including opinion and reactive emotion (the 'target' dimension). They all together vertebrate the conceptual model of Fake News that we present in the next section.

4 Conceptual Model

In this section, we present a Fake News conceptual model. It captures the main points of view identified through Section 3. The *Attacker sub-model* describes the context in which the Fake News is created. The *Target sub-model* allows designing the target population of the Fake News and the psychological effects generated by it. The *Fact sub-model* allows specifying the real fact on which a Fake News is based and the way it will be altered to produce the desired psychological effect.

To illustrate this section, we draw on an identified Fake News that was disseminated during the 2016 U.S. presidential campaign : it was published by a Donald Trump's supporter named Sean Hannity and stated that Trump helped 200 marines to come home after Iraq war as illustrated by the Figure 1.

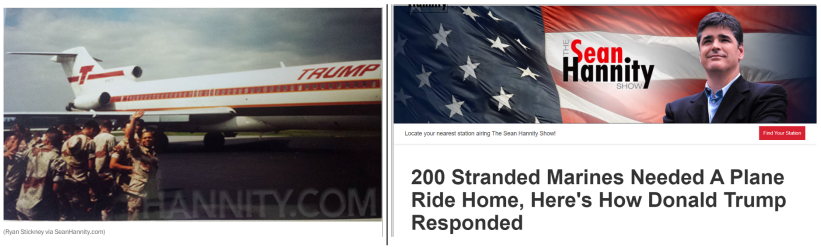


Fig. 1. Fake News stating that Donald Trump helped 200 Marines return to the US

4.1 Attacker sub-model

The Figure 2 is a part of the complete conceptual model describing the actors directly or indirectly involved in Fake News creation process. As mentioned above, it is common that the creation of a Fake News is the result of a structured disinformation campaign, in which we find a classic division into three levels of responsibility: strategic, operational and tactical. At the head of the organization

is the Strategic Attacker managing the Information Warfare and commanding Campaign Leaders. These Campaign Leaders manage the Disinformation Campaigns and have the Fake News Creator under their command to feed these campaigns with Fake News. The latter are created to be conform within a Disinformation Campaign. This is represented by the Is a part of relationship. Moreover, a Disinformation Campaign can also re-use a Fake News for its own purposes. Indeed, some of them can become viral and some campaigns could grasp the opportunity to re-use them to fulfill their own goals. It is represented by the Re-uses relationship. Both Information Warfare and Disinformation Campaign can be characterized by the context in which they are conducted. The Figure 3 illustrate how data are integrated into the different classes of the conceptual model.

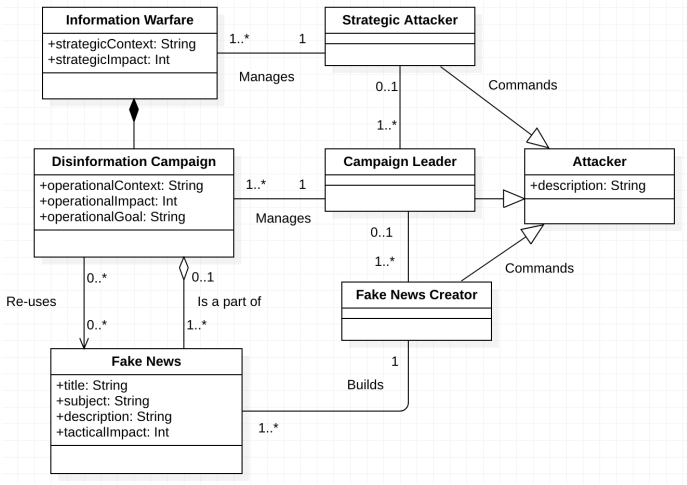


Fig. 2. Fake News Attacker Conceptual Model

4.2 Fact sub-model

This part of the model illustrated in Figure 4 details the facts on which a Fake News relies. Figure 5 shows how data populate it.

Fake News is the key notion. The entire diagram is created around it. title of a Fake News consists of its headline and is described by a string. subject corresponds to the category of knowledge it refers to. description is a textual summary of the key ideas contained in. tacticalImpact is an integer evaluating the number of potential readers, expressing the size of the targeted population.

A Fake News relies on at least one Real Fact. Since the context is important to understand the Real Fact itself, it is modeled through the context attribute. The description attribute provides a textual description of the Real Fact. It is

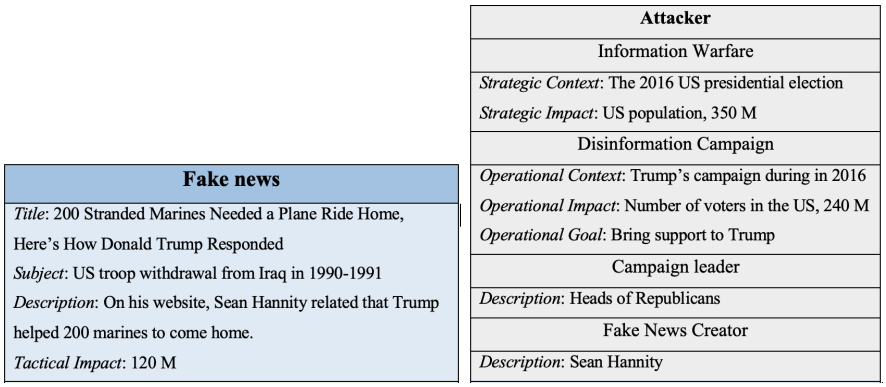


Fig. 3. Illustration of the use of the conceptual model Attacker part

characterized by a value stored in an enumeration : DATA, STATEMENT, or EVENT. A Real Fact can also be illustrated by a Document, which can either be a PICTURE, a FIGURE or a VIDEO.

False Facts contains its own description. It can be specified in three categories: made-up event, deformation of the truth or combination of true events, which are specializations of False Facts. Thus, False Facts are characterized as either eventDescription, deformationDescription or combinationType. When a false fact is a Combination of true events, it means that it refers to several unrelated Real Facts. Reality Distortion characterizes how strong the distortion is between the False Fact and the Real Fact it is based on. Its intensity can be LOW, MEDIUM or HIGH.

An Authority has a name and an expertise field. The three types of references to an authority discussed above make it possible to divide the concept into two simpler categories : true and false authorities. While the true authority did say or publish what is claimed to have been said or published, the false authority didn't. Thus, a Fake News can call upon an Authority, which can be either a True Authority, linked to a Real Fact, or a False Authority, linked to a False Fact. Both are specializations of Authority. In both cases, the Authority is characterized by a Credibility regarding the fact at hand. A False Authority also includes a Boolean attribute which models whether the Authority is a real entity or not.

4.3 Target sub-model

Fake News target specific communities and aim at affecting as many people as possible. To do that, they use cognitive mechanisms. The part of the model illustrated in Figure 6 deals with the target of the Fake News and the process through which it is influenced by the content, while Figure 7 shows a data example instantiation.

A Target corresponds to the group of people that the Fake News intends to reach and influence. targetCharacteristics are their descriptions. They determine

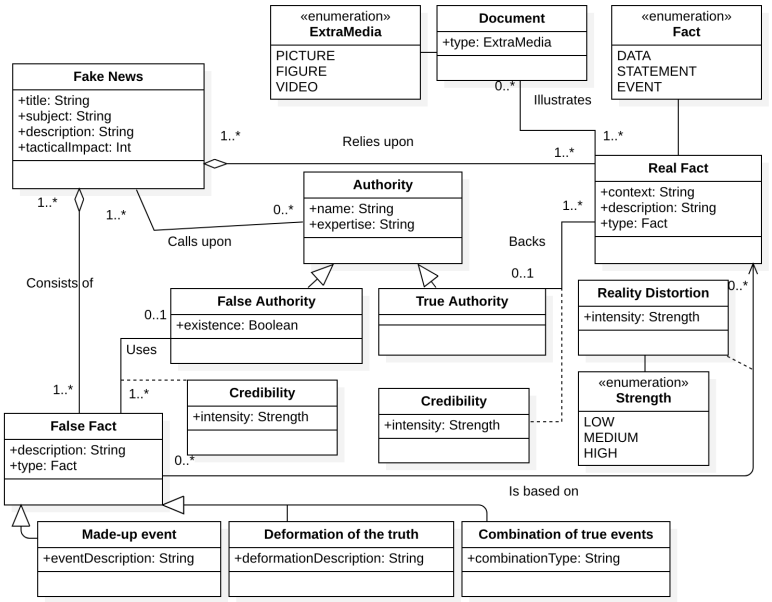


Fig. 4. Fake News Fact Conceptual Model

what makes them the Target. targetValues are ideas about which the readers will be especially sensitive about. An effective Fake News will twist these values to elaborate the Trigger Idea and maximize the Emotional Load. Fake News are spread through a media which can be either conventional, state-affiliated, alternative, or social. It is represented by Vector in the model. It also defines the Trust that the Target has in it. The latter is measured with an intensity : low, medium or high.

A Target expresses an Opinion about an opinionSubject. It is described with the opinionDescription and the type classifies the position: pro, anti, or neutral. This opinion depicts the target's original thinking before being influenced by the Fake News, i.e., changed or strengthened. Note that the manipulation of the Opinion starts with the Disinformation Campaign. The Goal of the Fake News can either be to WEAKEN or STRENGTHEN the opinion expressed by the target. Its action is determined by the Processed Information. Fake News contains Trigger Ideas which represent arguments in favor of the Processed Information. The latter is the mixing effect of an emotion and an information. It represents how the information is assimilated and understood by the target. It is directly responsible for influencing or changing the opinion of the target. Processed Information conveys an implicitConclusion through its Trigger Ideas. Fake News aims to influence opinions and lead the target to come to the desired conclusion. The emotional load enhances the phenomena. The trigger ignites an Emotional Load. Emotional load has a type which can be chosen among the eight standard emotions: JOY,

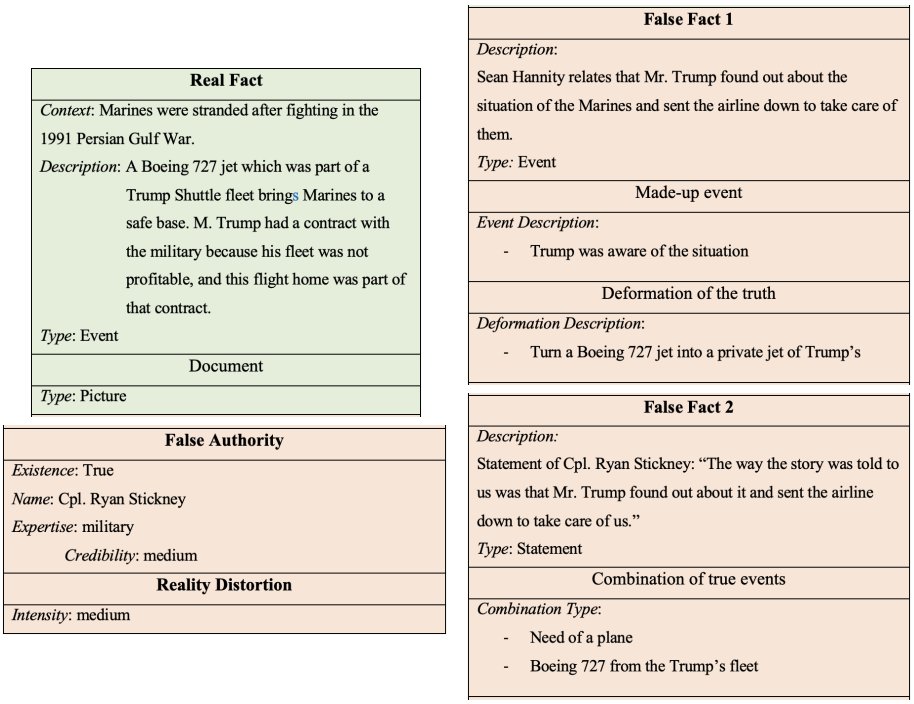


Fig. 5. Illustration of the use of the conceptual model Fact part

TRUST, FEAR, SURPRISE, SADNESS, ANTICIPATION, ANGER and DISGUST. The intensity of the load is evaluated based on the Plutchik wheel [16].

5 Discussion

The focus of our proposal is a Conceptual Model for Fake News. Our next further work includes a n intensive validation process, that we have started to accomplish through a preliminary validation phase with ten existing Fake News (see Table 2). Results have been very positive: with the support of the conceptual model, we have been able to characterize all of them in detail, what reinforces Fake News communication and management, our main purpose.

For lack of space, we will not detail the study, but we refer the reader to this website: https://people.irisa.fr/Nicolas.Belloir/public/ER2022. Some relevant conclusions have been obtained. Concerning the attacker part of the model, it is sometimes tricky to formally identify the operational and strategic levels in which a Fake News operates. This is not surprising since one of the pillars of disinformation is to hide the initiators to make it more credible. However, they can often be guessed. Concerning the Facts part of the model, we note that the different types of False Facts appear equitably. We also note that the stud-

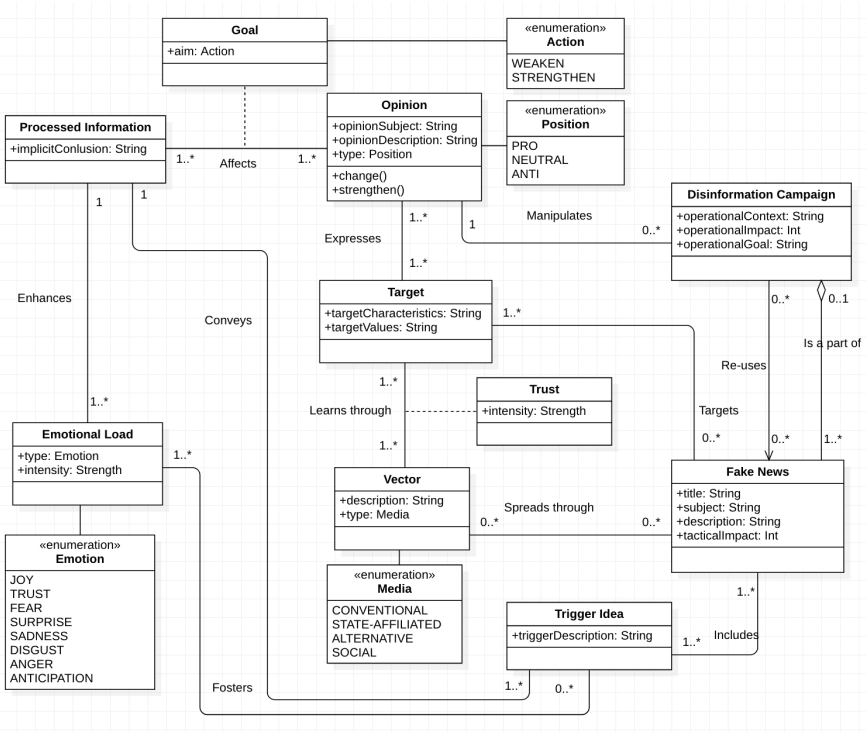


Fig. 6. Fake News Target Conceptual Model

Trigger Idea
Trigger description: participation of Trump during the Gulf War
Emotional Load
Type: Trust
Intensity: High

Vector
Description: published on Sean Hannity's website
Type: Alternative
Trust
Intensity: medium

Target
Target Characteristics: the US population
Target Values: voters
Opinion
Opinion subject: Trump
Opinion description: whether Trump is supported
Type: Pro
Processed Information
Implicit conclusion: Trump helped soldiers to come back home then he is a hero
Goal
Aim: Strengthen

Fig. 7. Illustration of the use of the conceptual model Target part

ied Fake News often play on the difference between correlation and causality, especially when a Fake News is built around several True Facts, which is not surprising. We also find an equal distribution of true and false authorities. Fi-

nally, if we look at the Target part, the emotional mechanisms are pretty easily identifiable, as well as the objective of the Fake News. However, if this study shows us that we manage to characterize existing Fake News with our model, the number of studied examples remains limited, and the conclusions we draw are to be considered with reserve. A more in-depth study will be necessary to consolidate the results and in the near future.

Num.	Fake News title	Description
1	Hillary has six months left	During the 2016 US presidential campaign, numerous rumours about Hillary Clinton's health were published. These rumours were posted after she fainted during the ceremony for the 9/11 victims in New York.
2	200 Stranded Marines Needed a Plane Ride Home, Here's How Donald Trump Responded	On his website, Sean Hannity related that Trump helped 200 marines to come home.
3	War Russia-NATO: An analyst from Pentagon foresees how it could end	An online article published in Ukraine claims that Russia could be easily defeated by NATO. The source is linked with Pentagon
4	Youngkin's false claim that McAuliffe 'opposes' election audits	Youngkin twitted a thread against his opponent Terry McAuliffe and claimed that the democrat was against audits.
5	Omar Holding Secret Fundraisers with Islamic groups tied to terror	Ilhan Omar was accused of having links with terrorists' groups during Minnesota Campaign
6	The Chinese President visits a mosque and asks Muslims to pray for the country to protect it from this disaster of Covid. It's now they discover Islam virtues!	The Chinese President asks Muslims to pray for the country to be protected against Covid-19
7	When did patient zero begin in US? [...]It might be the US army who brought the epidemic to Wuhan	The spokesperson of the Chinese Ministry of Foreign Affairs Lijian Zhao, insinuated in a tweet that the US Army actually brought Covid-19 to Wuhan. The fact that dead from Covid-19 might have been attributed to influenza is the proof that Covid-19 was in the US before the emergence in China.
8	Norway reclassifies Covid-19: No more dangerous than ordinary flue	The Norwegian Institute of Public Health, or NIPH, declared that Covid-19 had known several mutations and was now less dangerous. It is now no more dangerous than ordinary flue
9	Bill Gates backed polio vaccine disabled 47,000 kids	Bill Gates is accused of poisoning children in India with his polio vaccine
10	Here is what our brothers and sisters live every day all around the world because of the Gospel. Let us pray for our missionaries	Following the fall of Afghanistan, Christians are tortured and packaged in plastic bags until they die

Table 2. Fake News used for conceptual model evaluation

We also used the conceptual model to create a Fake News. The advantage of relying on a conceptual model is that it allows associating a process that will enable to control the Fake News generation. Such a formalized process is still a challenge, but our conceptual model is the first step to make it possible because it increases the interpretability of the resulting Fake News.

Indeed, explainability and interpretatbility is recognized as a major feature for future intelligent systems in AI and Machine Learning research. Known under the name eXplainable Artificial Intelligence (XAI), the aim is to produce intelligent systems that enhance the confidence of users to understand the underlying

reasoning and automations [9]. For Fake News context, to our knowledge, works involving explainability feature within Fake News detection methods are still at their beginning [17]. A promising line of work is to follow the XAI-based process proposed by [19] to facilitate building well-justified and explainable models for Fake News generation. The idea is to offer an approach that is understandable, trustable and manageable to humans, as suggested in [9]. The contribution of this paper establishes the foundation of such a process by solving the first, essential step of getting a shared understanding of what a Fake News is by introducing a precise conceptual model of Fake News. From that sound conceptual basis, the rest of the proposed XAI process can be applied in a reliable way. The explainability with our approach will be conceptually guided by the conceptual model which conforms the core of the contribution: to have an ontologically well-grounded definition of what a Fake News is, which is directly derived from the conceptual model.

6 Conclusion

This paper presents a conceptual model of Fake News allowing to identify and specify the relevant entities that conceptually characterize the different dimensions that must be considered in Fake News. We believe that this model will contribute significantly to improving tools for generating/detecting fake news. Under a sound conceptual modeling perspective, we argue that a correct Fake News management process will only be feasible if it is based on a precise conceptual modeling approach, as the one proposed in this paper. We have chosen to focus on the following main aspects: the production chain, the articulation around true and false facts, and the manipulation mechanism used by Fake News and based on emotions. We have confronted this conceptual model to up to ten different fake news in order to evaluate its conformity. The first results are encouraging. A larger study will be conducted in the near future in order to evaluate its scaling up. Moreover, the proposition addresses the first step of the XAI process proposed by [19], that it is crucial to make possible the full XAI-based process. The explainability with our approach is conceptually guided by the conceptual model which conforms the core of the contribution: to have an ontologically well-grounded definition of what a Fake News is, which is directly derived from the conceptual model. The next steps of the process are naturally our future work.

Acknowledgment

The authors would like to thank the final year engineering students of the Military Academy of St-Cyr Coëtquidan who worked on this project: Glenn Le Roux, Gaspard Croizat, Hugo Fouché, Émilien Frugier and Louis-Antoine Nicolazo De Barmon.

References

1. Allcott, H., Gentzkow, M.: Social media and fake news in the 2016 election. Journal of Economic Perspectives **31**(2), 211–36 (2017)
2. Ansar, W., Goswami, S.: Combating the menace: A survey on characterization and detection of fake news from a data science perspective. Inter. Journal of Information Management Data Insights **1**(2), 100052 (2021)
3. Arrieta, et al.: Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. Information Fusion **58**, 82–115 (2020)
4. Bani-Hani, A., Adedugbe, O., Benkhelifa, E., Majdalawieh, M.: Fandet Semantic Model: An OWL Ontology for Context-Based Fake News Detection on Social Media, pp. 91–125. Springer International Publishing (2022)
5. Belloir, N., Ouerdane, W., Pastor, O., Frugier, E., de Barmon, L.A.: A conceptual characterization of fake news: A positioning paper. In: Proceedings of the 16th International Conference on Research Challenges in Information Science (RCIS). Springer, LNBIP, Barcelona, Spain (2022)
6. Gilpin, L.H., Bau, D., Yuan, B.Z., Bajwa, A., Specter, M., Kagal, L.: Explaining explanations: An overview of interpretability of machine learning. In: IEEE 5th Int. Conf. on data science and advanced analytics. pp. 80–89 (2018)
7. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM computing surveys (CSUR) **51**(5), 93 (2019)
8. Gunning, D.: Explainable artificial intelligence (xai). DARPA **2** (2017)
9. Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., Yang, G.: Xai: Explainable artificial intelligence. Science Robotics **4**(37), 7120 (2019)
10. Katsaros, D., Stavropoulos, G., Papakostas, D.: Which machine learning paradigm for fake news detection? In: IEEE/WIC/ACM International Conference on Web Intelligence. pp. 383–387 (2019)
11. Kumar, S., Shah, N.: False information on web and social media: A survey (2018)
12. Lazer, D.M.J., Baum, M.A., Benkler, Y., Berinsky, A.J., Greenhill, K.M., Menczer, F., Metzger, M.J., Nyhan, B., Pennycook, G., Rothschild, D., Schudson, M., Sloman, S.A., Sunstein, C.R., Thorson, E.A., Watts, D.J., Zittrain, J.L.: The science of fake news. Science **359**(6380), 1094–1096 (2018)
13. Molina, M.D., Sundar, S.S., Le, T., Lee, D.: "fake news" is not simply false information: A concept explication and taxonomy of online content. American Behavioral Scientist **65**(2), 180–212 (2021)
14. N., W.S.Z.: Op-ed: Why can't a generation that grew up online spot the misinformation in front of them? Los Angeles Times, 6 November 2020. (2020), Available online: https://www.latimes.com/opinion/story/2020-11-06/colleges-students-recognizemisinformation (accessed on 11 January 2021)
15. Pierri, F., Ceri, S.: False news on social media: A data-driven survey. SIGMOD Rec. **48**(2), 18–27 (2019)
16. Plutchik, R.: Emotion: Theory, research, and experience. v.1: Theories of emotion (1980)
17. Shu, K., Cui, L., Wang, S., Lee, D., Liu, H.: Defend: Explainable fake news detection. In: Proc. of the 25th ACM SIGKDD. p. 395–405 (2019)
18. Shu, K., Sliva, A., Wang, S., Tang, J., Liu, H.: Fake news detection on social media: A data mining perspective. SIGKDD Explor. Newsl. **19**(1), 22–36 (2017)

19. Spreeuwenberg, S.: AIX: Artificial Intelligence needs eXplanation: Why and how transparency increases the success of AI solutions. LibRT BV, Amsterdam (2019)
20. Tandoc, E., Lim, Z., Ling, R.: Defining "Fake News": A typology of scholarly definitions. Digital Journalism **6**, 1–17 (2017)
21. Wang, C.: Fake news and related concepts: Definitions and recent research development. Contemporary Management Research **16**, 145–174 (2020)
22. Weiss, A.P., Alwan, A., Garcia, E.P., Kirakosian, A.T.: Toward a comprehensive model of fake news: A new approach to examine the creation and sharing of false information. Societies **11**(3) (2021)
23. Zafarani, R., Zhou, X., Shu, K., Liu, H.: Fake news research: Theories, detection strategies, and open problems. In: Proc. of the 25th ACM SIGKDD. pp. 3207—-3208. Association for Computing Machinery (2019)
24. Zhang, X., Ghorbani, A.: An overview of online fake news: Characterization, detection, and discussion. Information Processing & Management **57**(2), 102025 (2020)
25. Zhou, X., Zafarani, R.: A survey of fake news: Fundamental theories, detection methods, and opportunities. ACM Comput. Surv. **53**(5) (2020)

Model-Driven Engineering as the Interface for Tactical Operation Order of Mixed Robot/Human Platoons

Nicolas Belloir[1,2], Jérémy Buisson[1,2], and Lionel Touseau[1,2]

[1] Écoles de Saint-Cyr Coëtquidan, Guer, France
[2] IRISA, nicolas.belloir,jeremy.buisson,lionel.touseau@irisa.fr

Abstract. Technological advance is an enabler for the evolution of modern warfare for occidental armies. But the technological challenges are far beyond building new weapon systems such as semi-autonomous robotic systems and drones. An additional challenge is the elaboration of the necessary infrastructure substrate that will enable the smooth integration of these semi-autonomous systems into teamed human/robot platoons. In this paper, we explore how we can use Model-Driven Engineering (MDE - borrowed from software engineering) to address this specific challenge. We report our experiment on designing a suitable metamodel that reifies the concepts from the PROTERRE tasks of the French group to company units. The metamodel is then used as the abstract interface between the chiefs, their human subordinates, and their robots, each using their own modalities. Our preliminary results confirm the suitability of MDE technologies in this context. We also show that MDE adapts well to modalities that are unusual in software engineering, such as gesture communication.

Keywords: Model-Driven Engineering, Tactical operation order, Mixed platoon

1 Introduction

In network-centric warfare [1], holistic communications between field units and headquarters supposedly provide information and situation awareness superiority as an answer to Clausewitz's fog of war. The underlying assumption was: the better informed the commander is, the better and the timelier the plans and orders are, ensuring victory.

Technological advance contributes to the evolution of modern warfare beyond the network-centric concept. Nowadays, not only are the field units and headquarters connected by digital networks; but drones are also used as weapon systems and some digitized tools appear in headquarters. This trend is expected to go one step further with forthcoming weapon programs, e.g., [8]. In this context, not only the coordination of forces becomes more and more complex, but elaborating a consistent ecosystem, integrating semi-autonomous robotic systems

alongside human soldiers, across the battlefield and headquarters is a challenging task. Even the representation of operation plans and orders rises questions when we anticipate that orders will be shared and executed by teamed manned and unmanned systems of a single platoon.

With battlefield digitization, the pace of battle has already or will accelerate. As a result, reaction and decision-making times should be shortened. The way orders are delivered nowadays however cannot cope with such a fast pace. Although there have been some standardization efforts (e.g. [13]), orders still follow a verbose textual format. In addition, a robot, a drone, or any other semi-autonomous defense system, would not be able to understand oral or written instructions in natural language.

Based on this observation, it is necessary for leaders that orders can also be expressed in a machine-readable format. A model-based approach would allow to formalize the concepts related to tactical orders, by proposing a specific language for tactical orders, also called metamodel in Software Engineering.

Model-driven engineering (MDE) could help designing new interfaces to deliver modeled orders, based on this metamodel. Thanks to the abstraction layer provided by a metamodel-based approach, the same order could be delivered to machines as well as to human soldiers using an appropriate interface for each type of recipient.

Our contribution presented in this paper is two-fold. First a metamodel is proposed to express orders at the tactical level, relevant to both human and non-human units. Then multi-modal interfaces are built to deliver these orders, using the defined metamodel and model-driven engineering techniques. This approach would benefit leaders of mixed human and non-human platoons, and it would help them to adapt to the forthcoming changes in their profession.

Section 2 reports recent and prospective evolution to warfare. Then section 3 details the vision that we propose in this paper. Section 4 briefly introduces model-driven engineering. Section 5 presents the proposed metamodel and the results of our first experiment. Last, section 6 concludes the paper with a discussion.

2 Background

In his book [7], King discusses the evolution of the occidental armies at the beginning of the 21^{st} century. On the one side, several doctrinal and organizational changes were performed, including the shifts from citizen to professional, from single-service to joint, from national to multi-national. Operations become truly multi-domain, and therefore heterogeneous. On the other side, increasingly-secure digital communication technology enables interactions and cooperation across the larger and reconfigured battlefield. Introduction of information technology also adds the cyber domain to the battle space, in addition to land, air and maritime spaces. King observes that the division is the echelon of choice, identified in doctrines to deal with the perspective of high-intensity warfare, as

the divisional level is the one that has the capacity to coordinate multi-domain battle.

King also observes that, contrary to the initial expectations, the increased use of information technology did not lead to reduce the headquarter staff nor the number of command echelons. On the contrary, increased amount of information, increased range of operation and increased multi-domain cooperation yields to increased complexity at commanding at the division level. But the digital transformation of the headquarters and command posts, and more generally the digital transformation of warfare is not only about networking with field units to gather massively data, that can in turn be used to train some artificial intelligence to provide decision-making assistance as a response to the increased complexity at the headquarter. For instance, Mayorga *et al.* successfully experimented linear regression, naive Bayes and decision trees in the context of surveillance operations, to guide the conception of a military operation [9], depending on the location and expected results for the operation. The digital transformation also concerns field units beyond communication technology, like witnessed by projects for future combat systems, with the dawn of semi-autonomous robotic systems for the battlefield. Klare [8] reports three examples from the USA's perspective. The *SMET* (*Small Multipurpose Equipment Transport*) vehicle, while initially a robotic mule, is anticipated to evolve towards intelligence missions, then towards autonomously identifying and employing lethal weapons against the enemy. The *XQ-58A Valkyrie* is thought as an armed aerial drone, intended to clear the path for piloted aircrafts. The *Sea Hunter* project intends a similar purpose for naval operations, hunting for enemy submarines to assist manned warships. Other nations too race into this shift in warfare. European programs including *MGCS* (*Main Ground Combat System*) and *FCAS* (*Future Combat Aerial System*) programs, as well as the *TURMA* (*Teaming Unmanned Robotic Manned Architecture*) consortium also intend to team manned and unmanned vehicles such that robots and drones do fight under the supervision of human soldiers and commanders. According to Klare [8], Russia and China also have a similar agenda. Klare writes that even secondary powers develop such systems.

Klare [8] insists on the impact of these changes on the soldiers and on the commanders. We retain the two following ones. First, Klare points out the faster pace of combats thanks to the fact that fighting robots need no rest between battles. Continuous fighting is anticipated. Second, Klare notices the informational flood, that machines shall digest far faster than human commanders. Then there is a risk that human soldiers and commanders fail in their forthcoming new role of overseeing the drones and robots, and instead that the relationship between humans and robots gets reversed.

3 Vision

Integration of artificial intelligence is anticipated to increase the level of autonomy of the robots and drones accompanying fighting units on the field. We anticipate that AI will enable robots to achieve autonomously elementary ac-

tions, like those done by human soldiers at the lowest levels of the hierarchy, e.g., reconnaissance or support. So, in comparison to the current ones, the interface of future robots and drone will raise to a higher level of abstraction, comparable to lowest-level order languages.

Given that the robots shall be able to achieve similar actions to the ones made by human soldiers, it is therefore tempting that the chiefs interact with both in a similar manner. In addition, when the chiefs at the fire-team, squad or platoon levels give orders to their subordinate soldiers, they can employ various modalities, including voice, gesture, textual or graphical representation of the order.

So we rise the question whether the chiefs can abstract some details of their subordinates when giving orders. Like it will be described in section 4, model-driven engineering insists on the distinction between the abstract syntax of a language, and possibly-multiple concrete representations of the same information. Like described in section 5, the French PROTERRE [17] can play the role of the abstract syntax, and the modalities such as standardized gestures [17], STANAG 2014 [13]-like operation orders, APP-6 [2]-like overlay orders are possible concrete representations. Our vision is therefore to rely on model-driven engineering technology, such that the chiefs give their orders using any concrete modality at their convenience. In addition to direct communication, the orders shall be captured by the combat information system supporting the operation for broadcast to the subordinates. Having a machine-processable abstract syntax enables robots to receive the orders like human soldiers. And, possibly, the supporting combat information system shall adapt the order representation to allow the chiefs and their subordinates use different modalities. So, to some extent, in our vision, the abstract language plays the role of the interface between the chiefs and their subordinates, abstracting over whether the subordinates are human or robotic systems.

With this vision, one challenge is to design a suitable abstract language. Our method in this regard is that existing communication modalities such as PROTERRE gestures, operation orders and overlay orders provide some basis that can be abstracted to a metamodel. Then, in a second step in future work, we plan to adapt the resulting abstract language according to effective capabilities of robotic systems we use in our experiments.

4 Model-Driven Engineering

Model-Driven Engineering (MDE) [6, 11] is a family of technologies and methodologies that originate from object-oriented design in the field of software engineering. One of the key goals aimed by MDE is to make more systematic and more rigorous the documentation of the engineered software systems; and, at the same time, to enable software tools to manipulate this documentation. To achieve this goal, a key idea of MDE is to eliminate natural language to avoid any interpretation biases (that may result from cultural differences) and any ambiguities. Instead, MDE makes the documentation be expressed as a *model*,

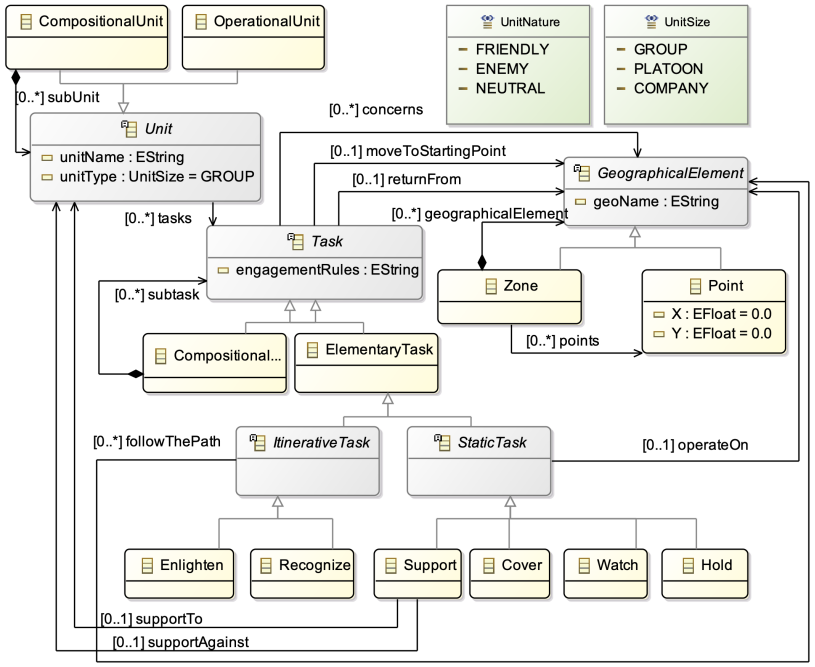
that relies on *concepts* described in a so-named *metamodel*. The model is an *instance* of the metamodel. Taking roots in object-oriented design, the model of a software system typically describes the *classes* of the *objects* manipulated by the modeled software system. The model is not restricted to structural documentation; it also allows the description of the functionalities, of the actors that interact with the software system, of the behavior at various levels of abstraction. Hence MDE supports the complete life-cycle of the software system. UML [18] is the most-often used metamodel when engineering a software system; SysML [16] when engineering a complex system or a system of systems. By construction, and in contrast with natural languages, modeling languages are such that models are easily manipulated by software tools, with the aim of computer-aided engineering.

Beyond software or complex system engineering, MDE has gone *domain-specific* [3, 4, 12]. For each domain, one may design a metamodel that describes the concepts used when engineering in that domain. The metamodel is itself an instance of a metametamodel such as MOF [10] or EMF's Ecore [14]. In fact, the metamodel is just a model whose domain is *modeling languages* and whose metamodel is the metametamodel. The stack of *meta* levels of modeling is conceptually indefinite, but usual metametamodels like MOF and EMF's Ecore are metacircular, i.e., they are instances of themselves, putting an end to the recursion.

The strength of this generic construction is that metametamodels come with software infrastructure, such as EMF for Ecore. Then this infrastructure comes with an ecosystem of tools, such as Sirius[3], Xtext[4], Acceleo[5], ATL [5], Henshin [15] to mention just of few of the EMF's ecosystem. From the metamodel, this infrastructure automatically generates infrastructure code compatible with itself, such that tools that manipulate models, built upon this infrastructure, can reuse the whole ecosystem off-the-shelf. A typical ecosystem provides model transformation engines and frameworks like ATL and Henshin, textual parser generators like Xtext, text generator engines like Acceleo, graphical editor generators like Sirius.

The above (very short) outline of the EMF ecosystem illustrates an additional characteristic of MDE that is worth being highlighted. The metamodel, by defining a modeling language, provides an *abstract syntax*, that is, it describes objects that shall appear in instance models in the form they are manipulated by the accompanying software tools. A metamodel engineer shall design multiple *concrete syntax* for a single metamodel, possibly using distinct modalities, including graphical notations and textual notations. By construction, concrete representations are no more than views of the (shared) model. Any modification made from one representation is reflected into the model, and therefore into all the other representations. Hence, mapping and synchronization between multiple representations are solved by construction. Said otherwise, the meta-

[3] https://www.eclipse.org/sirius/ (21/01/2021)
[4] https://www.eclipse.org/Xtext/ (21/01/2021)
[5] https://www.eclipse.org/acceleo/ (21/01/2021)



;

Fig. 1. Proterre French Army based Metamodel

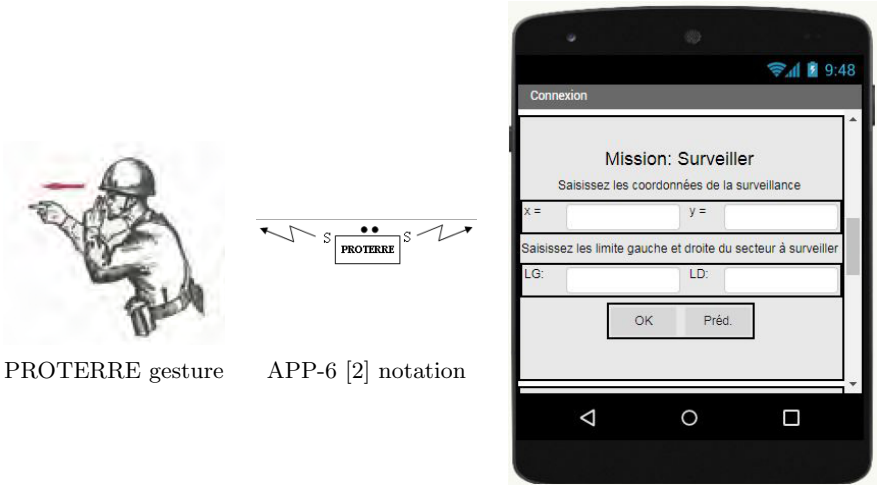
model describes some interfacing abstract language to be used for multi-modal communication between human and software systems.

For all the characteristics described in this section, MDE is a highly-relevant family of technologies to deal with the challenge of building an ecosystem, to support that idea of mixed human/robot platoons announced by forthcoming weapon systems.

5 Metamodel for Operation Order

In this section, we illustrate our vision by implementing part of the PROTERRE [17] approach into a metamodel. PROTERRE is the combat guide of the French Ground Army theorizing the main missions of a ground unit (from company size to group size). We choose PROTERRE due to our affiliation, but any similar combat guide from any other nation can be equally considered. In this section, we provide a metamodel as proof-of-concept. The metamodel doesn't represent the whole PROTERRE theory but it focuses on the relationship between units, tasks and geographical points as shown in figure 1.

A company is made up of 2 or 3 platoons. In a platoon, 3 or 4 groups can be organized. Depending on the level of responsibility, a chief may address an



PROTERRE gesture APP-6 [2] notation Tablet interface

Fig. 2. Ordering a *Watch* task of an area to a group of soldiers

order to a compositional unit or an operational unit. PROTERRE defines a set of well-identified missions. We call these missions *tasks* in the metamodel. Basically, tasks can be those defined as specialization of *ElementaryTasks*: *Enlighten*, *Recognize* .... All tasks refer to geographical elements. It can be a specific point, or specific areas (lines, circles, cities ...). For this paper, we are not trying to be exhaustive. All tasks are performed from a starting point to a return point. Some of the tasks are mostly static (for instance, observing a specific sector for the *Watch* task). Others are done by following a specific path (for instance, *Enlighten*). Some tasks are quite similar. *Enlighten* and *Recognize* mainly differ by the unit's reaction if it finds an enemy unit while performing the task. For the first one, the unit just points the enemy out. For the second one, it engages the enemy.

The metamodel is an abstract syntax. It can support one or more concrete syntax. For instance, ordering a *Watch* task for an area can result in different artifacts as illustrated by figure 2: the first part on the left shows the PROTERRE gesture, either for direct communication or using the IoT-like connected glove presented in figure 3. The location of the observers is the one occupied by the group of soldiers. The area to be monitored is by the direction of one hand. The type of task to be performed is done by the gesture of the other hand. The second part at the center shows the artifact to be used in graphical language or a diagram like the overlay order. Positions are be given by the position on the map for instance using APP-6 [2] symbols. The third part on the right shows the same order on a connected tablet. The group ID has already been defined.

The positions are given using geographic coordinates. The area to be monitored is given by azimuths.

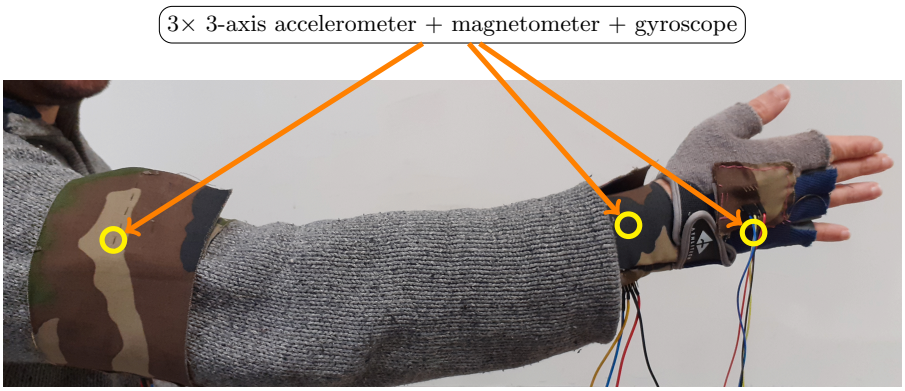


Fig. 3. A connected glove and armlet to capture arm gestures.

We carried out an experiment in order to illustrate the applicability of our proposal. We implemented a concrete syntax of the proposed metamodel into a tablet-based application. With this app, a chief is able to give some simple orders to a group of Lego Mindstorm robots, which are representative of typical ground semi-autonomous robots. Figure 4 illustrates our experimentation. A robot is selected using the app. The chief assigns a *Watch* task to a robot. The chief defines a place to perform the task using coordinates. The chief also defines two azimuths. When the order has been given, the robots move to the specified place, turn to be in the right position in order to look at the sector limited by the two azimuths. Figure 3 illustrates a simple connected glove we prototype, to use gesture recognition as a second concrete syntax. When the chief points her/his arm in a direction, the gesture is detected.

In future work, we will integrate all these objects as parts of a system of systems, such that the detection of the gesture will trigger issuing a *Watch* task. The glove sensors will be used to compute the azimuths to be used as parameters of the task. And the task will be sent to the robots, hence triggering the tactical action. The task will also be sent to soldiers' tablets, so the task will be translated on-the-fly from one concrete syntax to another one when appropriate. The latter is enabled by the use of the shared metamodel as the abstract syntax, at the interface between the chief, and the subordinate soldiers and robots.

6 Conclusion

In this paper, we summarized the recent and forthcoming evolution in occidental armies. The challenges raised by these changes encompass not only the production of new technologically-advanced equipment and organizational aspects, but

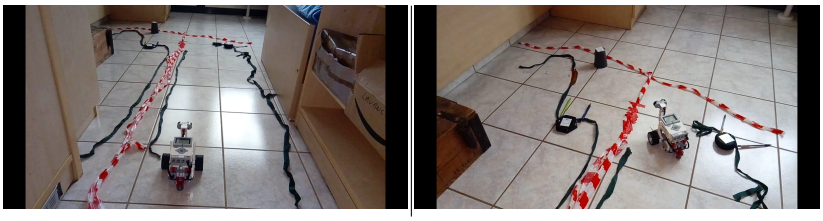


Fig. 4. Experimenting a "Watch" task given using a smartphone to a robot

also the elaboration of an ecosystem in which human soldiers and robots collaborate in mixed platoons. The vision that we defend in this paper is that Model-Driven Engineering (MDE), coming from the field of software engineering, is a candidate technology to provide the necessary infrastructure towards this ecosystem.

To illustrate the insights of the proposed vision, we reported our experiment in this direction. We presented a metamodel we elaborated to conceptualize the French PROTERRE tasks for group, platoon and company levels. Based on this metamodel, we developed a proof-of-concept to demonstrate the ability to use multiple modalities to issue orders to both human soldiers and robots. Still, the experiment we report is in early stage. We plan to investigate further integration of various concrete representations for orders, via multiple modalities, as well as mixing human soldiers and robots in the platoon.

With respect to traditional MDE in the context of software engineering, we enlarge the range of modalities for the concrete representations. MDE traditionally supports graphical and textual syntax. In our experiment, we already consider gestures as an additional modality.

References

1. Alberts, D.S., Garstka, J., Stein, F.P.: Network centric warfare: developing and leveraging information superiority. CCRP publication series. National Defense University Press, Washington, DC (1999)
2. APP-06 NATO Joint Military Symbology. Tech. Rep. NSO(JOINT)1231(2017)IERH/2019, NATO Standardization Office (2017). URL https://nso.nato.int/nso/nsdd/APdetails.html?APNo=1912&LA=EN
3. France, R., Rumpe, B.: Domain specific modeling. Software & Systems Modeling **4**(1), 1–3 (2005). DOI 10.1007/s10270-005-0078-1
4. Frank, U.: Domain-Specific Modeling Languages: Requirements Analysis and Design Guidelines. In: I. Reinhartz-Berger, A. Sturm, T. Clark, S. Cohen, J. Bettin (eds.) Domain Engineering: Product Lines, Languages, and Conceptual Models, pp. 133–157. Springer, Berlin, Heidelberg (2013). DOI 10.1007/978-3-642-36654-3_6
5. Jouault, F., Allilaire, F., Bézivin, J., Kurtev, I., Valduriez, P.: ATL: a QVT-like transformation language. In: Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications, OOPSLA

'06, pp. 719–720. Association for Computing Machinery, New York, NY, USA (2006). DOI 10.1145/1176617.1176691
6. Kent, S.: Model Driven Engineering. In: M. Butler, L. Petre, K. Sere (eds.) Integrated Formal Methods, Lecture Notes in Computer Science, pp. 286–298. Springer, Berlin, Heidelberg (2002). DOI 10.1007/3-540-47884-1_16
7. King, A.: Command: the twenty-first-century general. Cambridge University Press (2019)
8. Klare, M.T.: The Coming of Automated Warfare. Current History **119**(813), 9–14 (2020). DOI 10.1525/curh.2020.119.813.9
9. Mayorga, J., Borbúa, R.V., Reyes Ch., R.P., Gualotuña, T.: Jayor2: A Proposal of Information Management System for Command and Control Centers (C3i2) in the Armed Forces. In: A. Rocha, M. Paredes-Calderón, T. Guarda (eds.) Developments and Advances in Defense and Security, Smart Innovation, Systems and Technologies, pp. 271–280. Springer (2020). DOI 10.1007/978-981-15-4875-8_24
10. Meta Object Facility 2.4.2. Tech. Rep. formal/14-04-05, OMG (2014). URL https://www.omg.org/spec/MOF/2.4.2/
11. Schmidt, D.C.: Guest Editor's Introduction: Model-Driven Engineering. Computer **39**(2), 25–31 (2006). DOI 10.1109/MC.2006.58. Conference Name: Computer
12. Sprinkle, J., Mernik, M., Tolvanen, J.P., Spinellis, D.: Guest Editors' Introduction: What Kinds of Nails Need a Domain-Specific Hammer? IEEE Software **26**(4), 15–18 (2009). DOI 10.1109/MS.2009.92. Conference Name: IEEE Software
13. STANAG 2014: Formats for Orders and Designation of Timings, Locations and Boundaries. Tech. Rep. MAS(ARMY)0307-TOP/2014, NATO Military Agency for Standardization (2000)
14. Steinberg, D., Budinsky, F., Paternostro, M., Merks, E.: EMF: Eclipse Modeling Framework 2.0, 2nd edn. Addison-Wesley Professional (2009)
15. Strüber, D., Born, K., Gill, K.D., Groner, R., Kehrer, T., Ohrndorf, M., Tichy, M.: Henshin: A Usability-Focused Framework for EMF Model Transformation Development. In: J. de Lara, D. Plump (eds.) Graph Transformation, Lecture Notes in Computer Science, pp. 196–208. Springer International Publishing, Cham (2017). DOI 10.1007/978-3-319-61470-0_12
16. Systems Modeling Language version 1.5. Tech. Rep. formal/2017-05-01, OMG (2017). URL https://www.omg.org/spec/SysML/1.5
17. TTA 150: Le combat proterre en milieu ouvert. combat manual, French Ground Army (2014). In French
18. Unified Modeling Language version 2.5.1. Tech. Rep. formal/2017-12-05, OMG (2017). URL https://www.omg.org/spec/UML/2.5.1

Annexe C

Curriculum Vitæ détaillé

Nicolas BELLOIR	Maître de conférences hors classe (depuis septembre 2022)
Composante	UBS, détaché à l'Académie Militaire de Saint-Cyr Coëtquidan (AMSCC[5])
Laboratoire	IRISA/Archware et CReC[4]
section CNU	27 / Informatique

Vue synthétique

Responsabilités

Titulaire de la chaire de recherche « Cyberdéfense – Cybersécurité » St-Cyr Thales (AMSCC[5], Thales et Fondation St-Cyr). Cybersécurité du combat collaboratif, sécurité centrée sur la donnée. Depuis 01/2023.

Directeur adjoint du mastère spécialisé Cyberdéfense et Champs Immatériels (AMSCC[5]). Depuis 09/2022.

Positions précédentes

2018- ?	MCF – UBS, détaché à l'AMSCC	Chercheur IRISA/Archware et CReC
2017-2018	MCF – Université de Pau et des Pays de l'Adour (UPPA[1])	Chercheur IRISA/ Archware et CReC
2016-2017	MCF – UPPA[1], détaché à l'AMSCC	Chercheur IRISA/ Archware et CReC[4]
2005-2016	MCF – UPPA[1]	Chercheur LIUPPA[6]
2003-2005	ATER – UPPA[1]	Chercheur LIUPPA[6]
2001-2003	Doctorant Projet Européen Component+ (UPPA[1])	Chercheur LIUPPA[6]
1999-2001	Ingénieur d'étude - Transiciel	Informatique embarquée

Diplômes

2001-2004	Doctorat en Informatique de l'UPPA[1], direction F. Barbier et J-M Bruel
1998-1999	DEA Informatique de l'Image et du Langage - Université Paul Sabatier, Toulouse
1996-1998	Licence et Maîtrise en Informatique - UPPA[1]
1994-1996	DUT Informatique, Université Paul Sabatier, Toulouse

Parcours

Recruté en tant que **Maître de Conférences au département informatique de l'UFR Sciences et Techniques de l'UPPA**, j'effectue ma recherche au sein du

1. UPPA : Université de Pau et des Pays de l'Adour, Pau. http://www.univ-pau.fr/
2. Archware : ArchWare team/IRISA/Université de Bretagne Sud, Vannes. https://www-archware.irisa.fr/
3. IRISA : Institut de Recherche en Informatique et Systèmes Aléatoires. https://www.irisa.fr/
4. CReC : Centre de Recherche des écoles de St-Cyr Coëtquidan, Guer. https://www.st-cyr.terre.defense.gouv.fr/index.php/crec
5. AMSCC : Académie Militaire de Saint-Cyr Coëtquidan[5], Guer. https://www.st-cyr.terre.defense.gouv.fr/
6. LIUPPA : Laboratoire d'Informatique de l'Université de Pau et des Pays de l'Adour, Pau. http://liuppa.univ-pau.fr/
7. UBS : Université de Bretagne Sud. http://www.univ-ubs.fr

LIUPPA. Suite à des facteurs à la fois professionnels et personnels, je demande et obtiens une **mise en disponibilité d'un an pour m'occuper de mes jeunes enfants**. En 2012, mes activités de recherche évoluent vers un lien plus étroit entre ingénierie logicielle et ingénierie système. **Bien que restant rattaché au LIUPPA, je me retrouve isolé** et travaille en collaboration étroite avec l'équipe MACAO de l'IRIT à Toulouse. En septembre 2016, en recherche d'**un environnement de recherche plus favorable, j'effectue une mobilité**. Je déménage en Bretagne et j'intègre sur détachement l'AMSCC et son centre de recherche (CReC), ainsi que l'équipe Archware de l'IRISA (IRISA) sur la thématique de la cybersécurité et des systèmes de systèmes. **En septembre 2017**, l'UPPA **refuse de renouveler mon détachement** et je dois alors **réintégrer mes enseignements à Pau**, tout en assurant les enseignements prévus à l'AMSCC en vacations. En **mai 2018**, je **mute auprès de l'UBS** et suis immédiatement **placé en position de détachement à l'AMSCC**. En septembre 2022, je suis promu à la Hors-Classe. Depuis janvier 2023, je suis titulaire de la chaire de recherche « Cyberdéfense – Cybersécurité » St-Cyr Thales (AMSCC, Thales et Fondation St-Cyr).

Activités d'enseignement

Un auditoire scientifique

Mes enseignements se sont déroulés principalement dans le cadre de l'AMSCC (cursus d'ingénieur à l'École Spéciale Militaire (ESM) et cursus de licence à l'Ecole Militaire Inter-Armes (EMIA)) et de l'UFR S&T de l'UPPA. Le tableau suivant résume le nombre annuel moyen d'heures eq. TD d'enseignement.

Période	2005 à 2016	2016	**2017 à 2023**
moyenne eq. TD	232 h	204 h	**324 h** - pic à **408 h** en 2022/2023

Table C.1 – Nombre d'heures eq. TD moyen par période

Filière informatique :
- École d'ingénieurs – UBS/ENSIBS (depuis 2016)
- Licence Informatique – EMIA/AMSCC (depuis 2016)
- École d'ingénieurs – ESM/AMSCC (depuis 2016)
- Licence Informatique – UPPA (2004-2016 ; 2017-2018)
- Master Technologies de l'Internet – UPPA (2004-2016)
- DUT Informatique de Blagnac (2016)

Filière non informatique :
- Master GEII – Génie Elec. et Info. Industrielle - UPPA (2014-2016)
- Master MSID – Méthodes Stochastiques et Informatiques pour la Décision - UPPA (2006-2010)
- Master CFAO – Conception et Fabrication Assisté par Ordinateur – ENI de Tarbes (2006-2010)
- CPI – Classe Préparatoire Intégrée - UPPA (2010-2015)
- Formation continue ingénieurs - UPPA (2014-2016)

Des enseignements basés sur l'expérience académique et industrielle

- *Ingénierie logicielle et Système* : Introduction, UML, SysML, Ingénierie des modèles, Analyse de risque, Sécurité des systèmes.

— *Programmation procédurale, objet et mobile* : Java, C, C++, C#, initiation programmation mobile.
— *Système et programmation système* : programmation multi-processus et multi-threadées, Linux.
— *Cyber-sécurité* : méthodes d'ingénierie pour la cyber, EBIOS, sensibilisation à l'emprunte numérique.

Activités d'encadrement

Depuis 2021	Encadrement d'un **projet pédagogique** de 2e année de 3 à 4 élèves-ingénieurs. Le volume de travail est d'environ 80 h par élève	UBS/ENSIBS
2019	Encadrement d'un sous-officier candidat à un **diplôme d'ingénieur d'état** (IDPE). Accompagnement dans la rédaction du mémoire et dans la préparation de l'oral	AMSCC
Depuis 2018	Encadrement des activités de **Conduite de Projet** (CdP) de l'ESM. Projet mené par 3/4 élèves ingénieurs. Le volume de travail est d'environ 100 h par élève. Le travail de 2018-2019 a mené à une publication [Belloir et al., 2019] et celui de 2021-2022 a gagné le premier prix de la Journée des Sciences de l'AMSCC	AMSCC/ESM
Depuis 2018	Co-encadrement avec un MCF de sciences humaines et sociales (SHS) 2 **thèses professionnelles** de mastère (bac+6) par an. Ces thèses professionnelles sont en SHS ; j'amène de l'expertise technique aux élèves.	AMSCC/Master Cyber
2017	Encadrement un **projet tutoré** pour un groupe d'étudiants de première année du Master WMR	UBS
Depuis 2016	Tutorat et co-direction de **2 à 4 stages internationaux** de fin d'étude	AMSCC/ESM
2005-2016	Encadrement d'une douzaine de binômes d'étudiants de Master 1 et Master 2 Informatique pour la réalisation de leur **projet tutoré**	UPPA

Responsabilités pédagogiques

A l'AMSCC, la plupart des responsabilités est assurée par des officiers de l'Armée de Terre. Cela explique le faible nombre de responsabilités pédagogiques depuis que j'y suis en poste.

2023- ?	AMSCC : **Directeur du mastère spécialisé "Cyberdéfense et champs immatériels"**. Ce mastère forme des officiers en seconde partie de carrière ainsi que des cadres civils. Il vise à fournir des chefs capable de comprendre et de gérer une crise dans l'univers de la cyberdéfense mais également des champs immatériels. La particularité du mastère est que la formation est de 2/3 en sciences humaines et sociales et 1/3 en sciences de l'ingénieur. En lien avec mon co-directeur et notre adjoint administratif les missions sont les suivantes : recrutement, identification des intervenants, liens avec les Forces, liens avec l'Ecole des transmissions, organisation des exercices de crises, jurys, encadrement des thèses professionnelles, suivi des stages, ...
2022-2023	AMSCC : **Directeur adjoint du mastère spécialisé "Cyberdéfense et champs immatériels"**. Responsable de l'axe sciences de l'ingénieur...
2013-2015	UPPA : **Responsable de la seconde année du Master TI (UPPA)**. Suivi administratif du master, organisation et présidence des jurys et présidence de la commission de recrutement du master. Création d'une convention entre l'UPPA et l'EISTI, école d'ingénieurs privée. Permet aux étudiants de l'EISTI de suivre des modules du Master TI orientés recherche et de réaliser un stage recherche au sein du LIUPPA, ce qui leur confère un double diplôme. De leur coté, les étudiants du master pouvaient suivre des modules professionnalisant de l'EISTI
2012-2015	UPPA : **Membre du Comité de Perfectionnement du Master TI**
2007-2015	UPPA : **Responsable des stages pour le Master TI**. Prospection des sujets de stages et du contact avec les entreprises, de leur validation, de l'affectation des tuteurs de stages, du recueil des mémoires, de l'organisation des soutenances et, enfin, de tout le suivi administratif. La promotion pouvait aller de 20 à 30 étudiants
2007-2010	UPPA : **Responsabilité du Diplôme Universitaire STAGE**, délivré par l'UFR Sciences et Techniques de Pau. Ce DU permettait à des étudiants inscrits dans des formations n'offrant pas cette possibilité d'effectuer un stage de longue durée au sein d'une entreprise
2007-2016	UPPA : **Responsable des relations industrielles du département Informatique** : Création d'une plateforme de suivi des anciens étudiants, organisation d'un cycle de conférences animé par des anciens étudiants ou des intervenants industriels, organisation de rencontres annuelles étudiants-industriels-enseignants sous la forme d'un forum
2007-2016	UPPA : **Responsable des relations internationales** du département informatique. Montage d'échanges Erasmus, maintien des accords existants, gestion de l'administratif, suivi des étudiants à l'étranger

Activités de Recherche

Présentation générale

Thématiques de recherche :

— **Ingénierie dirigée par les modèles** (IDM) : métamodélisation, langages de modélisation (UML, SysML, DSML), méthodes

— **Cyber-sécurité** : Capture et préservation des propriétés de sécurité, méthodes d'assistance à l'architecte, modélisation des menaces, détection de la vulnérabilité hu-

maine, Fake News.

Type de systèmes applicatifs : Systèmes de systèmes, systèmes socio-techniques.

Domaine applicatifs : Défense

Collaborations industrielles :

— Depuis 2023 : Thales, **Titulaire de la chaire de recherche** "Cyberdéfense Cybersécurité"
— 2018-2023, Naval Group, bourse de thèse chaire Cyber Navale de l'Ecole Navale.
— Depuis 2017 : DGA, bourses de thèse via le Pôle d'Excellence Cyber.
— 2016, EDF R&D, bourse de stage master.
— 2008-2011, Néomades, Bidart, contrat Franck Barbier.

Collaborations transdisciplinaires : Je mène une collaboration active avec le Pôle Mutation des Conflits du CReC et le laboratoire Géode (essentiellement SHS) en cybersécurité : détection de la vulnérabilité humaine, génération/détection de Fake News, emprunte numérique du militaire, guerre cognitive ... En point d'orgue, en 2023 nous avons été mandatés pour fournir un rapport sur la guerre cognitive par le comité stratégique du service européen d'action extérieure (SEAE/EUEA), qui est le service diplomatique de l'Union Européenne.

Activité doctorale

Encadrements de thèses

Période	Doctorant	Taux	Rôle	Financement	Devenir
2023- ?	*Etienne Lemonnier*	40%	Dir.	Chaire "Cyberdéfense - Cybersécurité" St Cyr Thales	sans objet
2023- ?	*Jesús Antonio Sánchez Ramos*	33%	Co-Dir.	Pôle Excellence Cyber	sans objet
2022- ?	*Sidbewendin Angélique Yameogo*	33%	Enc.	Bourse UBS	sans objet
2018-2022	*Paul Perrotin*	33%	Enc.	Chaire Cyber des Systèmes Navals	Chercheur DGA
2016-2022	*Imane Cherfa*	50%	Enc.	Mi-temps en co-tutelle, Enseignante à l'Université de Blida 1, Algérie	MCF Université Blida 1
2017-2021	*Nan Messe*	33%	Enc.	Pôle Excellence Cyber	MCF, Université Toulouse - Jean Jaures
2009-2013	*Manzoor Ahmad*	25%	Enc.	Gouvernement Pakistan	Enseignant contractuel (CDI) - UPPA
2008-2011	*Youssef Ridene*	33%	Enc.	CIFRE Néomades	Senior Software Manager, Amazon Web Services
2007-2010 abandon	*Natacha Hoang*	50%	Enc.	Bourse agglo. paloise	Ingénieur d'études

Thèses soutenues

— **2008-2011**. ***Youssef Ridene***, "MATeL : A Domain-Specific Modelng Language for Mobile Phone Application Testing". (Encadrement avec Nadine Couture respectivement à 33% chacuns sous la direction de Franck Barbier). Financement Cifre avec la société Néomades. Soutenue en septembre 2011. Youssef a développé un environnement de test à distance pour téléphone portable utilisant une approche par métamodélisation.

— **2009-2013**. ***Manzoor Ahmad***, "Modeling and Verification of Functional and Non Functional Requirements of Ambient, Self Adaptive Systems". (Encadrement à 25% sous la direction de Jean-Michel Bruel se déroulant à distance à l'IUT de Blagnac). Financement bourse nationale du Pakistan. Soutenue en octobre 2013. Manzoor a proposé un processus de modélisation et de vérification des exigences fonctionnelles et non-fonctionnelles dans les systèmes auto-adaptatifs. Il a outillé cette approche en utilisant le langage SysML.

— **2017-2021**. ***Nan Zhang Messe***, "Sécurité par la conception : Une approche basée sur les assets pour réduire le fossé entre les architectes et les experts de sécurité". (Co-direction sous la responsabilité de Regis Fleurquin). Financement du Pôle d'Excellence Cyber en relation avec la DGA. Réalisée à l'UBS, la thèse a été soutenue le 7 janvier 2021. Nan a proposé une assistance à la détection de vulnérabilité dans les modèles d'architecture basés sur le concept d' "asset" et un processus pour formaliser la modélisation des menaces.

— **2016-2022**. ***Imane Cherfa***, "Mission Oriented Process for Systems of Systems Engineering". (Encadrement à 50% chacun sous la direction de Salah Sadou). Thèse en co-tutelle avec l'Université de Blida. La doctorante est enseignante à l'Université de Blida 1 et effectue sa thèse à mi-temps, a eu 3 enfants durant la période, ce qui explique sa durée. Elle a développée une approche de modélisation des Systèmes de Systèmes basée sur le paradigme de "mission".

— **2018-2022**. ***Paul Perrotin***, "Analyse de la vulnérabilité dans le cadre des systèmes de systèmes socio-techniques". (Encadrement à 33% chacun sous la direction de Salah Sadou et Antoine Beugnard). Thèse de la chaire de cyberdéfense des systèmes navals, en relation avec Naval Group, l'IMT-Atlantique et l'Ecole Navale. Le doctorant était basé à l'École Navale. Paul a proposé une approche de détection des vulnérabilités humaines dans les systèmes de systèmes socio-techniques.

Thèses en cours

— **2022- ?**. ***Sidbewendin Angélique Yameogo***. (Encadrement avec Wassila Ouerdane à 33% chacun sous la direction de Régis Fleurquin). Financement de thèse de l'Université de Bretagne Sud. La thèse, débutée en décembre 2022, vise à définir une approche d'explicabilité permettant de justifier, en utilisant un modèle conceptuel, pourquoi une nouvelle peut être qualifiée de fausse nouvelle.

— **2023- ?**. ***Jesús Antonio Sánchez Ramos***. (Encadrement avec Jamal El Hachem à 33% chacun sous la direction de Jérémy Buisson et ma co-direction). Financement du Pôle d'Excellence Cyber en relation avec la DGA. La thèse, débutée en janvier 2023, vise à intégrer les arbres d'attaque dans une démarche de détection des menaces au niveau architectural.

— **2023- ?**. ***Etienne Lemonnier***. (Direction Nicolas Belloir (40%), co-direction Jérémy Buisson, encadrement avec Jamal El Hachem et Lionel Touseau à 20% chacun). Financement de la chaire "Cyberdéfense - Cybersécurité St Cyr Thales". La thèse, débutée en octobre 2023, vise à définir une architecture type et un langage

d'architecture dédié pour la mise en place d'une approche centrée sur les données de type Data Centric Security, dans le cadre du combat collaboratif.

Thèses non soutenue

— **2017-2010**. ***Natacha Hoang***. (Encadrement à 50% sous la direction de Cong-Duc Pham). Thèse visant à développer un modèle de composants pour les réseaux de capteurs sans fil. Cette thèse visait à faire converger un sujet à la croisée des domaines de recherche des deux encadrants. La doctorante a abandonné pour raison personnelle. Elle est maintenant ingénieur d'étude dans une entreprise paloise.

Jury de thèse

— Co-encadrant : Paul Perrotin, IMT-Atlantique, décembre 2022.
— Co-encadrant : Imane Cherfa, Université de Blida 1, Algérie, mai 2022.
— Co-directeur : Nan Messe, Université de Bretagne Sud, janvier 2021.
— Examinateur : Alexandre Le Borgne, École des Mines d'Ales, janvier 2020.
— Co-encadrant : Manzoor Ahmad, Université de Toulouse, décembre 2013.

Diffusion et rayonnement

Mission d'expertise :

— 2022-2023 : Expert auprès du comité stratégique du Service européen d'action extérieure (SEAE/EUEA). Rédaction d'un rapport sur le thème de la Guerre Cognitive.
— 2016-2021 : Expert Crédit Impôt Recherche (CIR).

Animation et participation à des associations scientifiques nationales :

— 2013-2016 : **Membre du bureau de l'association SysML-France**. Nous avons monté et fait vivre cette association afin de disséminer l'utilisation du langage SysML puis plus largement de l'Ingénierie Système basée sur les Modèles auprès des industriels. Nous organisions des rencontres tous les 3 à 6 mois autour d'une question thématique. Nous réunissions entre une trentaine et une soixantaine de personnes à chaque fois.
— **Membre du GDR Génie de la Programmation et du Logiciel**.

Séminaires sur invitation

— 28 septembre 2023 : présentation lors du colloque "Intelligence artificielle et commandement militaire" organisé par Nexter et Naval Group.
— 20 Mars 2023 : présentation lors du colloque "Cybersécurité des grands événements publics" organisé par la chaire cybersécurité des grands événements publics.
— Séminaires à des industriels : Thales (2021), Airbus (2021), Naval Group (2020), Nexter (2020).
— 13 novembre 2019 : ProS, Universidad Politechnica de Valencia, Valencia, Espagne.
— 16 octobre 2019 : Equipe LATECE, UQAM, Montréal, Canada.
— 11 décembre 2016 : Journées Doctorants, Université de Blida I, Blida, Algérie

Participation à des comités de programmes

— Relecteur pour les journaux : Journal of Object Technology (2020- ?), Information and Software Technology (2015- ?) et Software and System Modeling (2012- ?)
— Membre du comité de programme du Doctoral Symposium de la conférence CAISE 2020
— Membre du comité de programme de la conférence ACS/IEEE International Conference on Computer Systems and Applications (AICCSA), 2015, 2016, 2017, 2018, 2019, 2020

— Membre du comité de programme de la conférence International Conference on Advanced Aspects of Software Engineering (ICAASE), (bi-annuelle) 2016, 2018
— Membre du comité de programme de la conférence INFORSID 2017, Toulouse, 2017
— Membre du comité de programme du Workshop on Models on Model-Driven Engineering for the Internet-of-Things (MDE4IoT) en 2018, 2019, 2020
— Membre du comité de programme de la conférence International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH) en 2014 et 2015
— Membre du comité de programme du International Workshop on Model Based Architecting and Construction of Embedded Systems (ACES-MB@Models) de 2008 à 2013
— Membre du comité de programme de la conférence International on Software Engineering and Knowledge Engineering (SEKE), 2009-2010

Participation à des comités d'organisation

— Journées du GDR GPL à Vannes en 2022
— 2^{nd} International Workshop on Security for and by Model-Driven Engineering (SecureMDE) dans le cadre de Models'20 (conférence de référence en ingénierie des modèles), octobre 2020.
— Journées SysML-France (février 2014, Toulouse et décembre 2014, Paris)
— Membre du comité d'organisation des conférences : 16^{ie} conférence sur les Langages et Modèles à Objets (LMO) / 6^{e} journées sur l'Ingénierie Dirigée par les Modèles (IDM) / 4^{e} Conférence francophone sur les Architectures Logicielles (CAL) / Journées du GDR GPL, Pau, 2010
— Membre du comité d'organisation de la conférence : IEEE/ACM MODELS, Toulouse, 2008
— Webmaster de la conférence : IEEE/ACM MODELS, Toulouse, 2008

Responsabilités collectives

— **Co-responsable de l'équipe** Architecture des Systèmes de Systèmes Sécurisés du CReC depuis 2020, composée de 3 permanents, 3 thésards et accueillant de nombreux stagiaires sous des formats divers (cadets étrangers, stagiaire de niveau M2 ...).
— **Président de la commission de sélection** d'un poste MCF - Université Toulouse (2021).
— **Vice-président de la commission de sélection** d'un poste MCF porté par l'UBS au profit d'un détachement à l'AMSCC en 2020.
— **Membre extérieur** à plusieurs **commissions de sélection d'enseignants sous contrats CDD LRU** au profit de l'UBS, cela depuis 2015.
— **Membre élu de la commission de spécialistes** de l'UPPA, section 27, de 2009 à 2011. Dans ce cadre, j'ai participé au recrutement de plusieurs postes de maîtres de conférences et de nombreux postes d'ATER.
— **Membre de la commission d'experts** pour l'informatique de l'UPPA en 2015 et 2016. Elle était en charge de l'étude des dossiers d'ATER et également d'aider le président à constituer les commissions de recrutement.
— Membre élu du **Conseil de Laboratoire du LIUPPA** de 2012 à 2016.

Annexe D

Bibliographie personnelle

Bibliographie personnelle

Nicolas Belloir

Articles en revue d'audience internationale à comité de lecture

[1] Imane Cherfa, **Nicolas Belloir**, Salah Sadou, Régis Fleurquin, and Djamal Bennouar. Systems of Systems: From Mission Definition to Architecture Description. *Systems Engineering*, 22(6):437–454, 2019. Q1/Q2 SCIMAGO.

[2] Manzoor Ahmad, **Nicolas Belloir**, and Jean-Michel Bruel. Modeling and Verification of Functional and Non-Functional Requirements of Ambient Self-Adaptive Systems. *Journal of Systems and Software*, 107(C):50–70, sep 2015. Core A.

[3] Eric Cariou, **Nicolas Belloir**, and Franck Barbier. OCL Contracts for the Verification of Model Transformations. In *The Pragmatics of OCL and Other Textual Specification Languages 2009*, volume 24, DENVER, CO, USA, 2009. Proceedings of the Workshop The Pragmatics of OCL and Other Textual Specification Languages at MoDELS 2009.

Articles en revue d'audience nationale à comité de lecture

[4] **Nicolas Belloir**, Jean-Michel Bruel, and Raphaël Faudou. Modélisation des exigences en uml/sysml. *revue Génie Logiciel. Numéro spécial Ingénierie des Exigences*, (111):6–12, December 2014.

[5] **Nicolas Belloir** and Jean-Michel Bruel. Développement basé composant : une approche centrée composition. *Méthodes Avancées de Développement des SI. Numéro spécial de la revue Ingénierie des Systèmes d'Information*, 10(6):59–80, 2005.

[6] **Nicolas Belloir**, Jean-Michel Bruel, and Franck Barbier. Intégration du test dans les composants logiciels. *Ingénierie des composants dans les systèmes d'information. Numéro spécial de la revue L'Objet*, 10(1):89–102, 2004.

Chapitres d'ouvrage d'audience internationale à comité de lecture

[7] **Nicolas Belloir**, Jérémy Buisson, and Lionel Touseau. Model-Driven Engineering as the Interface for Tactical Operation Order of Mixed Robot/Human Platoons. In Álvaro Rocha, Carlos Hernan Fajardo-Toro, and José María Riola Rodríguez, editors, *Developments and Advances in Defense and Security, proceedings of the 2021 Multidisciplinary International Conference of Research Applied to Defense and Security*, pages 205–214. Springer, 2022.

[8] Hans-Gerhard Groß and Colin Atkinson and Franck Barbier, **Nicolas Belloir**, and Jean-Michel Bruel. Built-in Contract Testing for Component-Based Development. In *Business Component-Based Software Engineering*, pages 65–82. Kluwer Academic Publishers, 2002.

Chapitres d'ouvrage d'audience nationale à comité de lecture

[9] Jean-Michel Bruel, Franck Barbier, **Nicolas Belloir**, and Fabien Roméo. Test de composants logiciels. In Mourad Oussalah, editor, *Ingénierie des Composants : Concepts, techniques et outils*, chapter 8, pages 229–245. Vuibert, June 2005.

Articles en conférence d'audience internationale avec comité de lecture

[10] **Nicolas Belloir**, Wassila Ouerdane, and Oscar Pastor. Characterizing Fake News: A Conceptual Modeling-based Approach. In Jolita Ralyté, Sharma Chakravarthy, Mukesh Mohania, Manfred A. Jeusfeld, and Kamalakar Karlapalem, editors, *Proceedings of the 41st International Conference on Conceptual Modeling (ER'22)*, pages 115–129, Cham, 2022. Lecture Notes in Computer Science, vol 13607. Springer. Core A.

[11] **Nicolas Belloir**, Wassila Ouerdane, Oscar Pastor, Émilien Frugier, and Louis-Antoine de Barmon. A Conceptual Characterization of Fake News: A Positioning Paper. In Renata Guizzardi, Jolita Ralyté, and Xavier Franch, editors, *Proceedings of the 16th Research Challenges in Information Science (RCIS'22)*, pages 662–669, Cham, 2022. Springer International Publishing. short paper, Core B.

[12] Paul Perrotin, **Nicolas Belloir**, Salah Sadou, David Hairion, and Antoine Beugnard. HoS-ML: Socio-Technical System ADL Dedicated to Human Vulnerability Identification. In *Proceedings of the 26th International Conference on Engineering of Complex Computer Systems (ICECCS'22)*, pages 1–6. IEEE, 2022. Short paper, Core A.

[13] Paul Perrotin, **Nicolas Belloir**, Salah Sadou, David Hairion, and Antoine Beugnard. Using the architecture of Socio-Technical System to analyse its vulnerability. In *Proceedings of the 17th Annual System of Systems Engineering Conference (SOSE'22)*, pages 361–366, 2022.

[14] Nan Messe, Vanea Chiprianov, **Nicolas Belloir**, Jamal El Hachem, Régis Fleurquin, and Salah Sadou. Asset-Oriented Threat Modeling. In *Proceedings of the 19th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom'20)*, pages 491–501. IEEE, December 29 - January 1 2020. Core A.

[15] Nan Messe, **Nicolas Belloir**, Vanea Chiprianov, Jamal El Hachem, Régis Fleurquin, and Salah Sadou. An Asset-Based Assistance for Secure by Design. In *Proceedings of the 27th Asia-Pacific Software Engineering Conference (APSEC'20)*, pages 178–187. IEEE-CS, 01-04 December 2020. Core B.

[16] Jérémy Buisson, **Nicolas Belloir**, and Jean Levrai Mbeck. Digitalization in Next Generation C2: Research Agenda from Model-Based Engineering Perspective. In *Proceedings of the 15th IEEE System of Systems Engineering Conference (SoSE'20)*, page 243–248. IEEE, 2020.

[17] Nan Messe, **Nicolas Belloir**, Vanea Chiprianov, Imane Cherfa, Régis Fleurquin, and Salah Sadou. Development of Secure Systems of Systems Needing a Rapid Development. In *Proceedings of the 14th IEEE System of Systems Engineering Conference (SoSE'19)*, pages 152–157. IEEE, 19-22 May 2019.

[18] **Nicolas Belloir**, Jérémy Buisson, and Olivier Bartheye. Metamodeling NATO Operation Orders: a proof-of-concept to deal with digitalization of the battlefield. In *Proceedings of the 14th IEEE System of Systems Engineering Conference (SoSE'19)*, pages 260–265. IEEE, 19-22 May 2019.

[19] Imane Cherfa, Salah Sadou, **Nicolas Belloir**, and Régis Fleurquin. Involving the Application Domain Expert in the Construction of Systems of Systems. In *Proceedings of the 13th System of Systems Engineering Conference (SoSE'18)*, pages 335–342. IEEE, 19-22 June 2018.

[20] Jörg Kienzle, Gunter Mussbacher, Omar Alam, Matthias Schöttle, **Nicolas Belloir**, Philippe Collet, Benoit Combemale, Julien Deantoni, Jacques Klein, and Bernhard Rumpe. VCU: The Three Dimensions of Reuse. In *Proceedings of the 15th International Conference on Software Reuse (ICSR'16)*, pages 122 – 137, Limassol, Cyprus, June 2016. Springer, LNCS.

[21] Manzoor Ahmad, Iulia Dragomir, Jean-Michel Bruel, Iulian Ober, and **Nicolas Belloir**. Early Analysis of Ambient Systems SYSML Properties using OMEGA2-IFx. In *Proceedings of the 3rd International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH'13)*, pages 147–154, Reykjavik, Iceland, 2013. SciTePress.

[22] **Nicolas Belloir** and Jean-Michel Bruel. Component Based Development : a Composition Oriented Approach. In *Proceedings of the IEEE 5th International Conference on Research, Innovation and Vision for the Futur (RIVF'07)*, pages 101–106. Universalis Publishing, 2007.

[23] **Nicolas Belloir**, Fabien Romeo, and Jean-Michel Bruel. Whole-Part based Composition Approach: a Case Study. In Magnus Larsson Ivica Crnkovic, editor, *Proceedings of the 30th Euromicro Conference on Component-Based Software Engineering*, pages 66–73, Rennes, France, September 2004. IEEE Computer Society Press.

[24] Franck Barbier and **Nicolas Belloir**. Component Behavior Prediction and Monitoring through Built-In Test. In *Proceedings of the 10th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS'2003)*, Huntsville, USA, 7-10 April 2003. IEEE Computer Society Press.

[25] **Nicolas Belloir**, Jean-Michel Bruel, and Franck Barbier. Whole-Part Relationships for Software Component Combination. In Gerhard Chroust and Christian Hofer, editors, *Proceedings of the 29th Euromicro Conference on Component-Based Software Engineering*, pages 86–91, Antalya - Turkey, September 2003. IEEE Computer Society Press.

[26] Franck Barbier, **Nicolas Belloir**, and Jean-Michel Bruel. Incorporation of Test Functionality into Software Components. In Hakan Erdogmus and Tao Weng, editors, *Proceedings of the 2nd International Conference on Commercial Off-The-Shelf (COTS)-Based Software Systems (ICCBSS'2003)*, pages 25–35, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[27] Christian Sallaberry and Eric Andonoff and **Nicolas Belloir**. WDBQS: A Unified Access to Distant Databases via a Simple Web Tool. In *Proceedings of the 11th International Conference on Databases and Expert Systems Applications 2000 (DEXA 2000)*, Lecture Notes in Computer Science 1873, pages 815–825, Kuwait, 4-8 September 2000. Springer.

[28] Eric Andonoff and Christian Sallaberry and **Nicolas Belloir**. WDBQL: A Web-Based Query Language for Remote Relational and Object-Oriented Databases. In *Proceedings*

of the 10th International Conference on Computing and Information'2000 (ICCI 2000), Lecture Notes in Computer Science, Kuwait, 18-21 November 2000. Springer.

Articles en atelier d'audience internationale avec comité de lecture

[29] Alexandre Le Borgne, **Nicolas Belloir**, Jean-Michel Bruel, and Thuy Nguyen. Formal Requirements Engineering for Smart Industries. In *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress - Workshop on Smart and Suistainable City*, pages 1028–1032. IEEE CPS, July 2016.

[30] **Nicolas Belloir**, Vanea Chiprianov, Manzoor Ahmad, Manuel Munier, Laurent Gallon, and Jean-Michel Bruel. Using Relax Operators into an MDE Security Requirement Elicitation Process for Systems of Systems. In *Proceedings of the 2014 European Conference on Software Architecture Workshops*, ECSAW'14, pages 32:1–32:4, New York, NY, USA, 2014. ACM.

[31] Manzoor Ahmad, João Araújo, **Nicolas Belloir**, Jean-Michel Bruel, Christophe Gnaho, Regine Laleau, and Farida Semmak. Self-Adaptive Systems Requirements Modelling: four related approaches comparison. In *Proceedings of the Comparing Requirements Modeling Approaches (CMA@RE'13) workshop, in the field of IEEE International Conference on Requirements Engineering 2013*, pages 37–42, Rio de Janeiro, Brasil, 16 July 2013. IEEE Computer Press.

[32] Youssef Ridène, **Nicolas Belloir**, Franck Barbier, and Nadine Couture. A DSML for Mobile Phone Applications Testing. In *proceedings of 10th Workshop on Domain-Specific Modeling in the field of SPLASH 2012*, pages 25–30, Reno, NV, USA, October 17–18 2010. ACM Press.

[33] Natacha Hoang, **Nicolas Belloir**, CongDuc Pham, and Sentilles Séverine. Valentine : a dynamic and adaptive operating system for wireless sensor networks. In *Proceedings of the 1st IEEE International Workshop on Component-Based Design of Ressource-Constrainted Systems (CORCS08) in the field of COMPSAC 2008*, pages 1297–1302, TurKu, Finland, Jul 28 - Aug 1 2008. IEEE Computer Society Press.

[34] **Nicolas Belloir**, Jean-Michel Bruel, and Franck Barbier. Component's Testability in Real-Time Systems. In *Proceedings of the 1st CARTS - Worshop on Advanced Real-Time technologies*, Aranjuez, Spain, 10-11 October 2002.

Articles en conférence ou atelier d'audience nationale avec comité de lecture

[35] **Nicolas Belloir**, Wassila Ouerdane, and Oscar Pastor. Caractérisation des fausses nouvelles – Une approche basée sur la modélisation conceptuelle. In *Actes du XXXXIeme congrès INFormatique des Organisations et Systemes d'Information et de Decision (INFORSID)*, pages 103–104, La Rochelle, May 2023.

[36] **Nicolas Belloir**, Wassila Ouerdane, and Oscar Pastor. Towards Security in a Dynamic Collaborative Operational Environment. In *Actes de la conférence C&ESAR 2022: Ensuring Trust in a Decentralized World*, Rennes, November 2022.

[37] Fernando Wanderley, **Nicolas Belloir**, Jean-Michel Bruel, Nabil Hameurlain, and João Araújo. Des buts à la modélisation système : une approche de modélisation des exigences centrée utilisateur. In *Actes du XXXIIeme congrès INFormatique des Organisations et Systemes d'Information et de Decision (INFORSID)*, pages 113–128, Lyon, 2014.

[38] Natacha Hoang, **Nicolas Belloir**, Xavier Detant, and Cong-Duc Pham. Un modèle de composant pour la reconfiguration dynamique de réseaux de capteurs sans fil. In *Actes de la 4ème Conférence francophone sur les Architectures Logicielles(CAL'10)*, pages 49–61, Pau, 2010.

[39] Eric Cariou, **Nicolas Belloir**, and Franck BarbierNidal Djemam. OCL contracts for the verification of model transformations (OCL 2009 workshop paper). In *Deuxièmes journées du GDR CNRS du Génie de la Programmation et du Logiciel*, March 2010.

[40] Eric Cariou, **Nicolas Belloir**, and Franck Barbier. Contrats de transformations pour la validation de raffinement de modèles. In *Actes des Journées sur l'Ingénierie Dirigée par les Modèles(IDM'09)*, Nancy, France, 25-26 March 2009.

[41] Jean-Michel Bruel, **Nicolas Belloir**, and Ahmad Manzoor. SPAS : un profil SysML pour les systèmes auto-adaptatifs. In *15ème Colloque National de la Recherche en IUT (CNRIUT), Lille, 08/06/09-10/06/09*, 2009.

[42] **Nicolas Belloir**, Jean-Michel Bruel, Natacha Hoang, and Cong-Duc Pham. Utilisation de SysML pour la modélisation des réseaux de capteurs. In *Actes de la conférence Langages et Modèles à Objets (LMO'08)*, pages 171–186, Montreal, Canada, 2-7 March 2008. RNTI.

[43] **Nicolas Belloir**, Jean-Michel Bruel, and Eric Cariou. Implémentation d'un modèle UML de composition hiérarchique. In *Actes de la conférence Langages et Modèles à Objets (LMO'07)*, pages 35–47, Toulouse, France, 27-29 March 2007. Hermès Sciences / Lavoisier.

[44] **Nicolas Belloir** and Fabien Romeo. Vérification a priori de modèle de composition logicielle. In *Actes du XXIIeme congrès Inforsid*, pages 163–178, 25-28 May 2004.

[45] **Nicolas Belloir**, Jean-Michel Bruel, and Franck Barbier. Application de la théorie de la relation Tout-Partie à la composition de composants logiciels. In *Actes du XXIeme congrès Inforsid*, pages 35–50, 3-6 June 2003.

[46] **Nicolas Belloir** and Fabien Roméo. Génération de composants testables avec testeur distribué. In *Actes du Workshop OCM-SI : Objets, Composants et Modèles dans l'Ingénierie des Systèmes d'Information*, Nancy, France, 3 June 2003.

[47] **Nicolas Belloir**, Jean-Michel Bruel, and Franck Barbier. Intégration du test dans les composants logiciels. In *Actes du Workshop OCM-SI : Objets, Composants et Modèles dans l'Ingénierie des Systèmes d'Information*, Nantes, France, 4 June 2002.

[48] **Nicolas Belloir** and Jean-Michel Bruel. Intégration du test dans les composants logiciels, 13 December 2001. Actes de la journée de travail des Ggroupes 3.1 et 3.2 - GDR I3.

[49] **Nicolas Belloir**, Jean-Michel Bruel, and Franck Barbier. Formalisation de la relation Tout-Partie : application à l'assemblage des composants logiciels. In *Actes des Journées Composants : flexibilité du système au langage*, Besançon, France, 25-26 October 2001.

Tutoriels en conférence d'audience internationale avec comité de lecture

[50] **Nicolas Belloir** and Jean-Michel Bruel. Model-Based System Engineering with SysML, 29-31 July 2013. Half-day tutorial shared between SIMULTECH and ICSOFT 2013.

Mémoires valant pour graduation et rapports techniques

[51] Kevin Limonier, **Nicolas Belloir**, Didier Danet, Saïd Haddad, Julien Nocetti, and Stéphane Taillat. Cognitive Security : from FIMI threats analysis to cognitive security policy. Technical report, European Union External Action/STRATCOM, 2023.

[52] **Nicolas Belloir**. *Composition logicielle basée sur la relation Tout-Partie*. PhD thesis, Université de Pau et des Pays de l'Adour, December 2004.

[53] **Nicolas Belloir**. Une approche navigationnelle pour l'interrogation de bases de données via le Web. Master's thesis, Université Paul Sabatier, 1999.

Titre : Un voyage dans les modèles : du logiciel aux systèmes de systèmes socio-techniques.

L'utilisation de modèles basés sur des langages graphiques est désormais usuelle lors de la conception de systèmes. Ces langages, associés à des processus, permettent de définir des méthodes d'ingénierie. On parle d'ingénierie dirigée par les modèles (IDM). Nous nous inscrivons ici dans cette dynamique, moins en tant que contributeurs de l'IDM qu'en tant qu'utilisateurs. Nous défendons l'idée selon laquelle les systèmes modernes, en raison des liens étroits entre les problématiques humaines, logicielles et techniques, peuvent être considérés à travers une seule et même vision : les systèmes de systèmes socio-techniques. Ainsi, il est possible de croiser les vues en une approche intégrée plus complète. Trois grands domaines sont abordés.

L'utilisation de modèles basés sur des langages graphiques est désormais usuelle lors de la conception de systèmes. Ces langages, associés à des processus, permettent de définir des méthodes d'ingénierie. On parle d'ingénierie dirigée par les modèles (IDM). Nous nous inscrivons ici dans cette dynamique, moins en tant que contributeurs de l'IDM qu'en tant qu'utilisateurs. Nous défendons l'idée selon laquelle les systèmes modernes, en raison des liens étroits entre les problématiques humaines, logicielles et techniques, peuvent être considérés à travers une seule et même vision : les systèmes de systèmes socio-techniques. Ainsi, il est possible de croiser les vues en une approche intégrée plus complète. Trois grands domaines sont abordés.

Title : A Journey into Models: From Software to Socio-Technical Systems of Systems.

Keywords : Model Driven Engineering, Socio-technical System of Systems, Secure by design

Using models described by graphical languages is now common in system design. These languages, coupled with processes, define engineering methods, known as Model-Driven Engineering (MDE). We align with this trend, not as MDE contributors but as users. We advocate that modern systems, due to close links between human, software, and technical issues, can be viewed through a unified lens: socio-technical systems of systems. Thus, a more comprehensive integrated approach is feasible. Three major areas are addressed.

Firstly, in the context of general specification and design of systems of systems, we propose an approach to formalize flexibility in requirements. Additionally, we suggest linking mission engineering with architecture specification to establish a stable intermediate model over time.

Secondly, a focus is placed on security by design. We propose an approach to enhance collaboration between domain experts and security experts by reifying the concept of an "asset." We then present a method to identify the human vulnerabilities within a socio-technical system-of-systems.

Finally, we examine the applicability of MDE in the Defense domain, initially through the concept of "order". Then we explore model usage in the realm of cyber influence warfare.