Titre: Title:	Engin d'explication d'anomalies sur le bus ARINC 429 basé sur les ontologies
Auteur: Author:	Nuno-Gonçalo Silva-Pinto
Date:	2023
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Silva-Pinto, N.-G. (2023). Engin d'explication d'anomalies sur le bus ARINC 429 basé sur les ontologies [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/53448/

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Engin d'Explication d'Anomalies sur le Bus ARINC 429 Basé sur les Ontologies

NUNO-GONÇALO SILVA-PINTO
Département de génie informatique et génie logiciel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie informatique

Mai 2023

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Engin d'Explication d'Anomalies sur le Bus ARINC 429 Basé sur les Ontologies

présenté par **Nuno-Gonçalo SILVA-PINTO**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Omar ABDUL WAHAB, président
Gabriela NICOLESCU, membre et directrice de recherche
Michel GAGNON, membre et codirecteur de recherche
Amal ZOUAQ, membre

DÉDICACES

À ma famille et mes amies,
À mes professeurs, mes mentors et mes collègues de travail,
Aux licornes, aux glorieux émissaires et aux patnès de la piscine Calixa qui ont cru en moi,
Merci pour tout votre support !

REMERCIEMENTS

Je remercie chaleureusement ma directrice de recherche, Prof. Gabriela Nicolescu, et mon co-directeur, Prof. Michel Gagnon, pour leur support et pour m'avoir rattrapé quand il le fallait. Cette recherche n'aurait pas été possible sans l'aide de Bombardier inc. Je voudrais remercier toute l'équipe et spécialement Rémi Benito qui m'a offert son soutien durant nos rencontres. Les conseils offerts par Collins et par le pilote Hans Oba m'ont aussi grandement aidé dans ma recherche. Enfin, je voudrais aussi exprimer mes sincères remerciements à Jean-Simon, Mikaela, Jean-Yves, Jean-Guy Felipe et Amine pour ces bons moments passés dans le laboratoire et pour leur support moral dans l'accomplissement de mon projet de recherche.

RÉSUMÉ

La cybersécurité des avions est un secteur d'activité en plein essor. Pour s'assurer de protéger les communications de l'avionique, il est important que les canaux soient sécuritaires. Toutefois, le standard de bus ARINC 429, communément présent chez les avions commerciaux, n'a pas intégré dans son architecture la cybersécurité. De plus, la plupart des contre-mesures proposées, notamment les Système de détection d'intrusion (IDS) n'ont pas pris en compte les attaques ne ciblant que la couche d'application logique ou la chaîne d'approvisionnement dans leur conception.

Notre objectif de recherche est d'étudier la viabilité de l'explication d'anomalies pour les parties prenantes de l'avionique par le biais d'une ontologie modélisant le standard de bus ARINC 429. Cette ontologie, qui constitue le coeur de notre contribution, a été intégrée à ATMONTO, une ontologie de gestion du trafic aérien produit par la NASA et à l'ontologie MITRE D3FEND qui permet de prendre en compte les concepts de cybersécurité.

Le cadre MITRE ATT&CK Entreprise produit deux scénarios d'attaques qui sont importés dans notre ontologie. Un cas de *spoofing* qui intègre une composante avionique malveillante dans l'avion et un deuxième cas qui fait intervenir un déni de service à travers une attaque sur la chaîne d'approvisionnement. La création de ces scénarios est évaluée par un expert spécialisé dans le web sémantique ainsi que par un pilote professionnel. Nous établissons alors un Engin d'Explication d'Anomalies (EEA) qui nous permet d'expliquer aux parties prenantes une anomalie détectée. Cela se traduit par la conception de questions de compétences qui sont basées sur un haut niveau de logique de détection. Le prototype comprend aussi une base de connaissances construite à partir de l'ontologie développée.

Notre engin a fait l'objet de 3 cas d'études : (1) un radioaltimètre voyou entrant en compétition avec la redondance, (2)un EFB trafiqué durant la chaîne d'approvisionnement qui affiche une valeur de vitesse erronée et (3) un pilote automatique voyou qui se désactive à une étape de vol critique.

Pour pouvoir tester les attaques sophistiquées et les différentes règles de détection, nous démontrons dans la méthodologie que la base de connaissance réussit les tests de cohérence et répond à des questions de compétences pertinentes. Les résultats sont présentés pour ces cas. On fournit ainsi une capacité adéquate pour informer le pilote, l'ouvrier de maintenance ou le développeur d'IDS de l'attaque qui se déroule. Une contre-mesure de la situation est aussi offerte. Les résultats s'appuient sur des tests de cohérences et des scénarios proposés par des experts du domaine.

Le travail réalisé permettra, dans le futur, d'intégrer d'autres types de bus avioniques, comme l'AFDX ou le MIL-STD-1553, de produire des données de vol à partir d'un simulateur de vol et d'incorporer des techniques en traitement automatique du langage naturel (NLP) pour enrichir le pont humain-machine de l'engin.

ABSTRACT

Cyber attacks targeting physical devices soars in avionics. To ensure the protection, channels need to be secure. However, the ARINC 429 bus standard, commonly used in commercial aircraft, does not incorporate cybersecurity into its architecture. Furthermore, most proposed countermeasures, including Intrusion Detection Systems (IDS), have not taken into account attacks targeting the logical application layer or the supply chain in their design.

Our objective in this research is to investigate the viability of explaining anomalies to stakeholders in avionics through an ontology that models the ARINC 429 bus standard. This ontology, which forms the core of our contribution, has been integrated into ATMONTO, an air traffic management ontology produced by NASA, and the MITRE D3FEND ontology, which incorporates cybersecurity concepts.

The MITRE ATT&CK Enterprise framework produces two attack scenarios that are imported into our ontology. One scenario involves spoofing, integrating a malicious avionics component into the aircraft, while the second scenario involves a denial of service attack through an attack on the supply chain. The creation of these scenarios is evaluated by an expert specialized in semantic web and a pilot. We establish an Explanation of Anomaly Engine (EEA) that allows us to explain a detected anomaly to stakeholders. This involves designing competency-based questions that are based on a high level of detection logic. The prototype also includes a knowledge base built from the developed ontology.

Our engine has been subjected to three case studies: a rogue radio altimeter competing with redundancy, a tampered Electronic Flight Bag (EFB) in the supply chain displaying incorrect speed value, and a rogue autopilot that disables itself at a critical stage of flight.

To test more sophisticated attacks and various detection rules, we demonstrate in the methodology that the knowledge base passes coherence tests and provides relevant competency-based answers. The results are presented for these cases, providing adequate capability to inform the pilot, maintenance worker, or IDS developer of the ongoing attack. A countermeasure to the situation is also provided. The results are based on coherence tests and scenarios proposed by domain experts.

The work accomplished will enable the future integration of other types of avionic buses, such as AFDX or MIL-STD-1553, the generation of flight data from a flight simulator, and the incorporation of Natural Language Processing (NLP) techniques to enhance the human-machine interface of the engine.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

AAO	Avionics Analytics Ontology
AFCS	Automatic Flight Control System
AFDX	Avionics Full-Duplex Switched Ethernet
AHRS	Attitude and Heading Reference System
AID	Aircraft Interface Devices
AISD	Airline Information Services Domain
AMI	Avionique modulaire intégrée
ARINC	Aeronautical Radio Incorporated
ATO	Air Traffic Organization
ATS	Service de circulation aérienne
ATC	Contrôle de la circulation aérienne
ATM	Gestion du trafic aérien
ATMONTO	Air Traffic Management Ontology
COTS	commercial sur étagère
CMU	Communication Management Unit
DOS	Déni de service
EASA	European Aviation Safety Agency
EFB	Electronic Flight Bag
EEA	Engin d'explication d'anomalies
FAA	Federal Administration Aviation
FLS	Field Loadable Software
FMS	Système de gestion de vol
GAO	Government Accountability Office
ICD	Interface Conrol Document
IDS	Système de détection d'intrusion
IMA	Integrated Modular Avionics
IATA	International Air Transport Association
IETF	Internet Engineering Task Force
ICS	Document de Contrôle d'Interface
LSAP	Loadable Software Airplane Parts
LRU	Line-Replaceable Unit
MAC	Code d'authentification de messages
NASA	National Aeronautics and Space Administration

NextGen	Next Generation Air Transportation System
NLP	traitement automatique du langage naturel
NOTAM	Notices to Airman
OACI	Organisation de l'aviation civile internationale
OWL	Web Ontology Language
PFD	Primary Flight Display
RDF	Resource Description Framework
SAMOD	Simplified Agile Methodology for Ontology Development
SAO	Situation Awareness Ontology
SESAR	Single European Sky ATM Research
TCAS	système de contrôle des trajectoires et de l'évitement de collisions
TTP	Techniques Tactiques et Procédures
UCO	Unified Cybersecurity Ontology
UML	Unified Modeling Language
W3C	World Wide Web Consortium
XAI	Explainable Artificial Intelligence

LISTE DES ANNEXES

CHAPITRE 1 INTRODUCTION

1.1 Contexte

La sécurité, leitmotiv de la règlementation du contrôle aérien, étudie la protection contre les cas de blessures intentionnelles. Aux annexes 17 et 19 de la Convention de Chicago, l'Organisation de l'aviation civile internationale (OACI) stipule de facto, à l'article 44, qu'un de ses principaux objectifs est de promouvoir la SÉCURITÉ de vol dans la navigation aérienne internationale [1]. Après un détournement ou un accident d'avion, la scène internationale révise de fond en comble les normes de l'aviation. C'est ce qui s'est produit après les attentats du 11 septembre 2001. L'OACI a alors mis en place des mesures pour contrer l'évolution de pratiques criminelles. Ces dernières décennies, ce sont les cyberattaques qui sont en essor. La Government Accountability Office (GAO) stipule qu'il est primordial que la Federal Administration Aviation (FAA) renforce la surveillance et la protection contre les risques de cybercriminalité liés à l'avionique [2]. Même si lancer une attaque informatique dans un avion demeure très complexe, avec une faible probabilité de réussite, la seule réussite d'un attaquant aurait, pour le domaine, un impact non négligeable. C'est pour cette raison que les cyberattaques sont catégorisées comme la deuxième plus grosse menace chez les compagnies aériennes, après les désastres naturels [3].

1.2 Éléments de la problématique

Les avions modernes sont de plus en plus informatisés avec des systèmes analogiques convertis en systèmes digitaux. Une tendance actuelle est celle de l'automatisation. L'objectif ne consiste pas seulement à rendre le vol plus sûr, mais aussi à éliminer les erreurs humaines. Malgré ces prouesses, les communautés de l'aviation argumentent qu'on ouvre la porte à de nouveaux types d'erreurs et de vulnérabilités opérationnelles en particulier en ne faisant que trop ou pas assez confiance envers l'automatisation [4, 5]. Si convaincre un pilote que la machine est là pour le soutenir n'est pas une tâche aisée, un abus de confiance de l'automatisation se révèle aussi nuisible pour le pilote. Ce dernier peut finir par omettre de vérifier que toute l'information nécessaire est disponible et qu'elle est de qualité acceptable.

Incorporer la cybersécurité dans les avions nécessite des tâches cognitives hautement sophistiquées. Une collaboration entre le pilote et l'avion grâce aux communications des systèmes de bus depuis les commandes de vol électriques (ou *fly-by-wire*) assure l'automatisation de l'avionique.

À travers les systèmes des bus, le protocole le plus communément utilisé aujourd'hui dans les vols commerciaux est le standard ARINC 429. Toutefois, ce protocole, instauré depuis plus de 30 ans, n'a pas considéré la sécurité dans ses objectifs [6] et n'a reçu aucune évaluation de sécurité publique [7]. L'ARINC 429 serait ainsi vraisemblablement peu résilient face aux attaques informatiques [6]. De plus, le bus a reçu une attention significative ces dernières années à cause du manque d'authentification [7]. Par ailleurs, même le protocole de bus moderne AFDX emprunte l'architecture archaïque de l'ARINC 429. Du fait de la grande portée qu'a toujours l'ARINC 429, il s'agit ainsi d'un protocole intéressant pour de potentiels criminels.

Peu de mesures de mitigation existent pour les standards avioniques de développement de logiciels (DO-178) et de matériels informatiques (DO-254). Le mieux qu'un ingénieur puisse faire pour son avion c'est de limiter sa connectivité avec l'Internet, la dépendance de plusieurs sous-systèmes à un seul logiciel et l'usage d'un produit commercial off-the-shelf (COTS) (emprunté de l'expression anglaise *commercial of-the-shelf*) [8]. Durant les récentes années, des mesures plus efficaces pour sécuriser l'avionique ont commencé à être étudiées. L'attention est surtout portée sur la sécurité à bas niveau par l'analyse de signal ou de paquets. La couche logique a toutefois souvent été omise, ce qui peut berner les IDS [9]. Cette problématique repose sur l'hétérogénéité des composantes avioniques qui nécessitent des contre-mesures à grande échelle. Les composantes peuvent en effet provenir de différents fabricants ou être basées sur des technologies différentes. Il est ainsi nécessaire que les contre-mesures au domaine hautement standardisé de l'aviation respectent les normes et les réglementations du secteur aéronautique.

Une ontologie devient prometteuse pour uniformiser le vocabulaire utilisé dans le domaine de l'avionique. Elle permet d'identifier et distinguer des sous-domaines de l'aviation, dans la gestion du trafic aérien (ATM) et dans la sécurité des aéroports [10, 11]. Un langage commun favorise la standardisation de nouvelles terminologies et des concepts, pour ainsi faciliter l'intégration des connaissances de ces domaines [12]. Les autorités de l'aviation se demandent encore si d'autres sous-domaines ne pourraient pas bénéficier également de cette technologie.

C'est en considérant ces caractéristiques que nous choisissons comme solution de développer un Engin d'explication d'anomalies (EEA) basé sur des ontologies. L'expérience combine trois domaines de connaissances : le trafic aérien avec Air Traffic Management Ontology (ATMONTO) [13], la cybersécurité avec MITRE D3FEND [14] et le bus avionique de l'ARINC 429 *onto-by-429*. Cette dernière ontologie a été produite au sein de notre expérience et constitue le cœur de ce mémoire auquel nous lions le standard de bus ARINC 429 avec

les deux autres domaines de connaissances mentionnées pour produire une ontologie interdisciplinaire *onto-by-wire*. L'outil s'appuie sur une ontologie interopérable pour répondre à des questions liées aux incidents. Il indique alors des contre-mesures tout en s'appuyant sur des règles de logique traçables et consistantes. En d'autres mots, lorsqu'une question lui est posée, ce dernier y répond et déclare des erreurs de logiques lorsque celles-ci surviennent.

Pour l'engin, on définira une anomalie comme étant un paquet avionique qui ne se situe pas dans les bornes de valeurs attendues transmises ou reçues par un composant n'ayant subi aucune altération malicieuse. À l'inverse, un paquet situé dans les bornes normales se définit comme étant une normalité.

Si différentes taxonomies d'attaque sur l'avionique ont déjà été réalisées, leur logique et leur crédibilité demeurent douteuses puisqu'elles s'inspirent souvent d'une analyse de risque sans plan d'architecture reconnu [15]. La classification des termes peuvent n'obéir à aucun standard et semer des quiproquos. Le développement de scénarios d'attaques et l'évaluation basé sur le profil d'un attaquant est ainsi produit en utilisant le cadre reconnu MITRE ATT&CK [16]. De plus, pour éviter d'offrir une analyse générique, on choisit d'attaquer le problème en adoptant, dans ce mémoire, une méthodologie adéquate tout en justifiant notre environnement de conception. En modélisant un domaine interconnecté, on optimise la collaboration en terme de cybersécurité des parties prenantes tels un pilote, un ouvrier de maintenance ou un ingénieur de conception.

1.3 Défis de recherche

Plusieurs recherches, dont la nôtre, se heurtent au problème d'obtenir des ressources de la part de fabricants et de partenaires. Malgré le signal d'alarme de la GAO, l'idée que le partage des connaissances puisse engendrer des brèches demeure répandue dans le milieu. Nous sommes conscients que divulguer des secrets présente des risques pour la sécurité de l'aviation et qu'il est pertinent d'éviter que des pirates informatiques découvrent comment un système opère. Toutefois, nous argumentons que cette mesure basée uniquement sur l'offuscation est nuisible à long terme. Au lieu de ralentir un individu malveillant, elle sert plutôt à dissimuler les vulnérabilités déjà présentes dans l'avion.

Un autre défi repose sur le peu de littérature lié à notre sujet de recherche. Les vulnérabilités de l'avionique n'ont surtout été étudiées que depuis cette dernière décennie. Il est donc difficile de trouver des sources pertinentes et fiables. Leur industrie et le monde de la recherche deviennent victimes de contraintes et d'un sérieux manque de connaissances. Notre solution se doit d'être axée sur l'échange de ressources mutuelles de base de données d'attaques,

d'allocation pour le testage ou de conception même d'un banc d'essai à source ouverte [17]. De surcroit, la quantité limitée de littérature nous offre peu d'inspiration pour déterminer la contribution la plus judicieuse à entreprendre afin de gérer les attentes des parties prenantes. Sous une différente optique, ceci nous laisse une voie plus libre dans la contribution que nous souhaitons apporter.

Un troisième défi repose sur la validité scientifique de nos données. Pour assurer un tel objectif, il est important pour nous d'éviter le biais de l'expérimentateur. Étant à la fois en charge de la conception et la réalisation de l'expérience en plus de l'interprétation des résultats, on veut éviter que nos préjugés personnels affectent notre travail. Pour éviter ce biais, nous appliquerons des méthodologies standardisées, le plus de variables indépendantes possible et des expériences en double aveugle.

1.4 Objectifs de recherche

Nous prenons en considération ces défis et répondons à la question de recherche suivante :

Est-ce qu'une ontologie intégrant l'avionique et la cybersécurité peut expliquer correctement des anomalies présentes dans le système de bus ARINC 429 aux parties prenantes ?

Le principal objectif de cette recherche est de développer une EEA basé sur des ontologies. La solution proposée respecte la méthodologie de conception SAMOD [18] et offre des résultats testables. Étant donné la difficulté d'obtenir des données réelles sur l'ARINC 429, notre expérimentation prend la forme d'une preuve de concept qui pourra, dans une recherche ultérieure, utiliser de véritables données. Quant à celles liées au trafic aérien, elles proviennent de vols disponibles par la National Aeronautics and Space Administration (NASA) [13]. D'une part, la satisfaction des résultats obtenus est vérifiée grâce à des tests d'inconsistances et des réponses aux questions de compétences. D'autre part, l'ontologie est validée lors du processus itératif par un ingénieur spécialisé dans le Web sémantique ainsi qu'avec un pilote professionnel. En résumé, nous ciblons l'explication efficace aux parties prenantes d'attaques informatiques et de contre-mesures, le tout avec un outil résultant d'un effort de standardisation et d'interopérabilité.

Dans le but de développer cet EEA, nous définissons les sous-objectifs suivants :

a) Analyser le modèle de menace possible sur le réseau de bus ARINC 429.
b) Appliquer les tactiques d'un adversaire sur la base de MITRE ATT& CK basé sur des observations du monde réel.

c) Concevoir une ontologie qui modélise le domaine de connaissance de l'aviation, l'avionique et la cybersécurité.
d) Vérifier et valider notre application avec des tests et des autorités pertinentes.

La contribution de cette recherche constitue, somme toute, en un prototype d'une preuve de concept de l'Engin d'Explication d'Anomalies basé sur les ontologies pour le standard ARINC 429. La méthodologie offre un apport au domaine hétérogène de l'avionique puisqu'elle vise un cadre standardisé dans son approche. Cette approche se veut également interdisciplinaire en intégrant l'avionique, la gestion de vol et la cybersécurité. Enfin, les cas d'études proposés font intervenir une évaluation du *modus operandi* des attaquants les plus aptes à cibler l'avionique.

1.5 Plan du mémoire

Ce mémoire se divise de la manière suivante :

Chapitre 2, Tour d'horizon des concepts de bases : Les concepts de l'avionique, des standards de bus, de la cybersécurité et de l'ontologie sont expliqués ;
Chapitre 3, Revue de littérature : Survol de l'état de l'art des bancs de tests de l'ARINC 429, des mesures offensives et défensives sur les bus et de l'ontologie dans les domaines de l'aviation et de la sécurité ;
Chapitre 4, Revue des méthodologies : Survol des méthodologies pertinentes en vue de la réalisation de notre objectif de recherche ;
Chapitre 5, Une nouvelle approche d'Engin d'Explication d'Anomalies : Une description du développement de l'explorateur d'anomalies basé sur les ontologies ;
Chapitre 6, Résultats : Les méthodes d'expérimentations et les résultats collectés utilisant l'engin d'explication d'anomalies, et ;
Chapitre 7, Conclusions : Conclusions, limites et futures approches.

CHAPITRE 2 TOUR D'HORIZON DES CONCEPTS DE BASES

Dans ce chapitre, nous abordons les concepts de bases jugés pertinents pour implémenter notre EEA.

2.1 Les systèmes avioniques

2.1.1 Les parties de l'avion

L'augmentation de la densité des vols est rendue possible aujourd'hui grâce à la sophistication transparente de l'avionique et des systèmes de contrôle aérien de la navigation dans les postes de pilotage.

La gestion du trafic aérien se fait au sol avec des radars, dans les airs avec des capteurs et dans l'espace avec le GPS. La participation des organismes de standardisation ces dernières décennies, comme NextGen et SESAR, a généré un développement standardisé de l'avionique. Tel qu'illustré à la figure 2.1, les divers équipements avioniques offrent aide, contrôle et prise de décisions pour l'équipage durant toutes les phases de vol. On retrouve, notamment le système de contrôle des trajectoires et de l'évitement de collisions (TCAS), un instrument de bord qui permet l'évitement de collisions. Un autre exemple est celui du pilote automatique qui réduit la charge de travail des pilotes pour les longs vols. On réduit ainsi le niveau de prises de conscience de la situation (ou *situational awareness*).

Les cinq éléments clefs de l'avionique comprennent : les équipements de navigation, les bus de données et leur câblage, les commandes de vol électriques, la puissance de l'appareil et enfin l'Integrated Modular Avionics (IMA) qui décrit le réseau distribué en temps réel à bord de l'avion (voir Figure 2.2). On abordera surtout dans cette recherche les trois premiers aspects [19, 20].

2.1.2 Les équipements de navigation

Le système de gestion de vol, ou *Flight Management System* (FMS)

Les avions modernes ont augmenté leur apport en automatisation pour réduire la charge de travail de l'équipage et du contrôleur aérien. Utilisés à bon escient et avec un équipage formé, les avions civils modernes ont souvent plus recours à un navigateur qu'un ingénieur de vol. De même, certains systèmes avioniques plus avancés ont retiré un des pilotes du poste de pilotage [22]. Le Système de gestion de vol (FMS) inclut un logiciel d'automatisation qui se

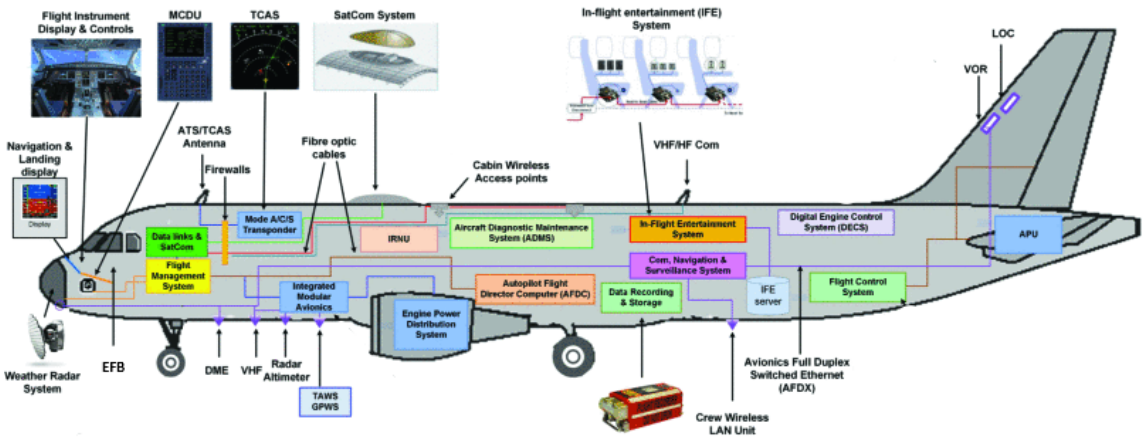


FIGURE 2.1 Avioniques, et modèle simplifié des systèmes de gestion de l'information et des interconnexions de l'avion [19].

connecte aux autres systèmes. Il facilite la gestion du vol et du poste de pilotage. Il assiste aussi le pilote pendant son vol en fournissant des renseignements sur le pilotage, la navigation, les estimations ou la consommation de carburant. Avec une base de données, des points de cheminement (*waypoint*) et des paramètres d'instructions pour le pilote automatique, l'avion est apte à mettre le cap vers des points appropriés, mettre à jour le contrôle des moteurs ou changer la vitesse et l'altitude [23].

La sacoche de vol électronique, ou *Electronic Flight Bag* (EFB)

La sacoche de vol électronique (EFB) est un outil qui renferme des applications assistant le pilote et son copilote et stockant des données de vol. L'EFB prend en compte la météo, les fuseaux horaires et le contrôle du trafic aérien. Optionnel dans certains vols, on l'utilise quand même fréquemment pour automatiser le FMS. Elle réalise entre autres les calculs de planification de vols et l'affichage de documents d'aide à la navigation. Des tableaux de données peuvent être chargés vers l'avionique à travers des disques, une prise USB ou un PC de diagnostics. La communication se fait enfin au moyen des commandes de vol électriques et de standards comme l'ARINC 429 ou l'AFDX. On retrouve enfin des EFBs intégrés au système et des EFBs portatifs qu'un pilote peut transporter dans ses mains en dehors de l'avion [23].

Aircraft Communications Addressing and Reporting System (ACARS)

L'ACARS est un autre système avionique servant à envoyer des messages textuels entre le sol, les aérodromes et l'aéronef. Il utilise les ondes radio VHF ou SATCOM ainsi qu'un

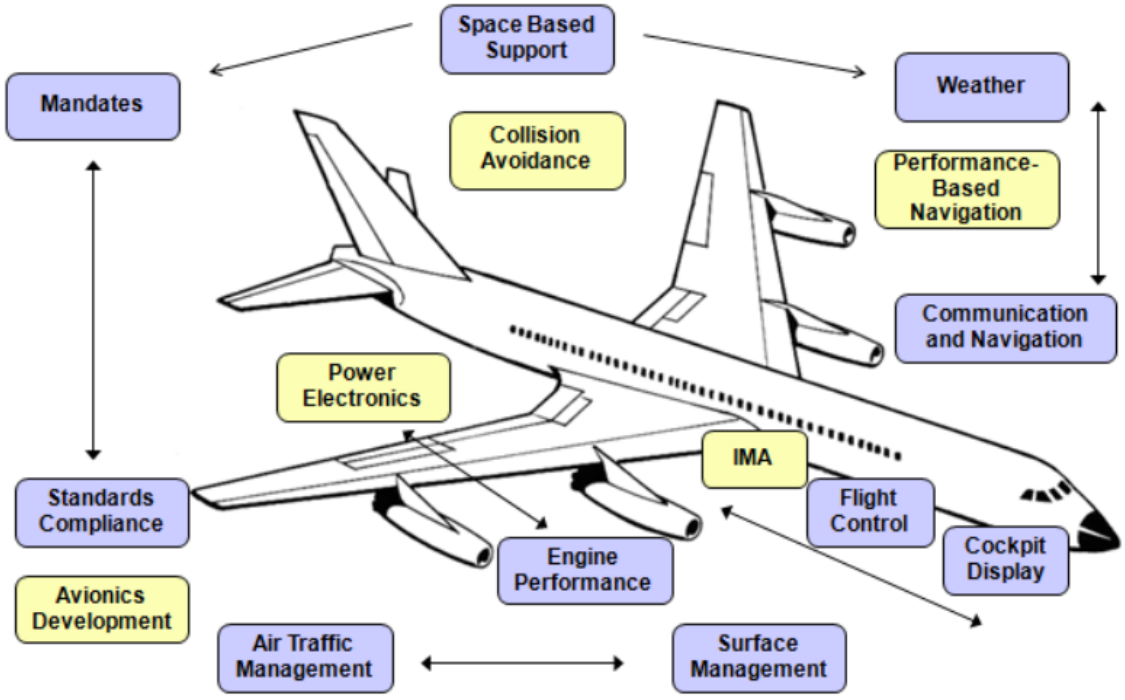


FIGURE 2.2 Résumé des techniques avioniques [21].

réseau au sol pour transmettre ses messages. Cet engin de liaison de données achemine les communications des opérations entre le pilote, la tour de contrôle, la maintenance et les autres parties prenantes. Il permet de garder un avion efficace et sûr. Le système ne comporte toutefois aucune mesure de chiffrement. Un individu peut tout à fait intercepter un message ACARS, ce qui compromet son authentification et sa confidentialité [24].

Field loadable software (FLS)

Les mises à jour sont nécessaires dans un aéronef. Elles se font le plus souvent avec les FLS, via un CD-ROM ou un correctif téléchargeable en ligne. Le logiciel doit être intensément testé puisqu'une seule défaillance peut mettre en danger la vie des passagers. C'est pourquoi des tests de consistances sont régulièrement faits, que ce soit au niveau de l'interface, des paramètres d'alimentation ou du câblage [25].

Automatic Flight Control System (AFCS)

L'Automatic Flight Control System (AFCS) intègre les fonctions de contrôle du tangage de l'autopilote et de l'automanette (*autothrottle*) tels les trajectoires de vols longitudinaux et les contrôles de vitesse. Il est composé de capteurs et d'ordinateurs qui surveillent en permanence l'état de l'avion. Ces états comprennent sa vitesse, son altitude et sa trajectoire. En diminuant la charge de travail de pilote dans sa gestion du vol, ils contribuent à améliorer la sécurité, la fiabilité et l'efficacité des opérations aériennes [26]. La figure 2.3 présente le schéma d'un AFCS conventionnel.

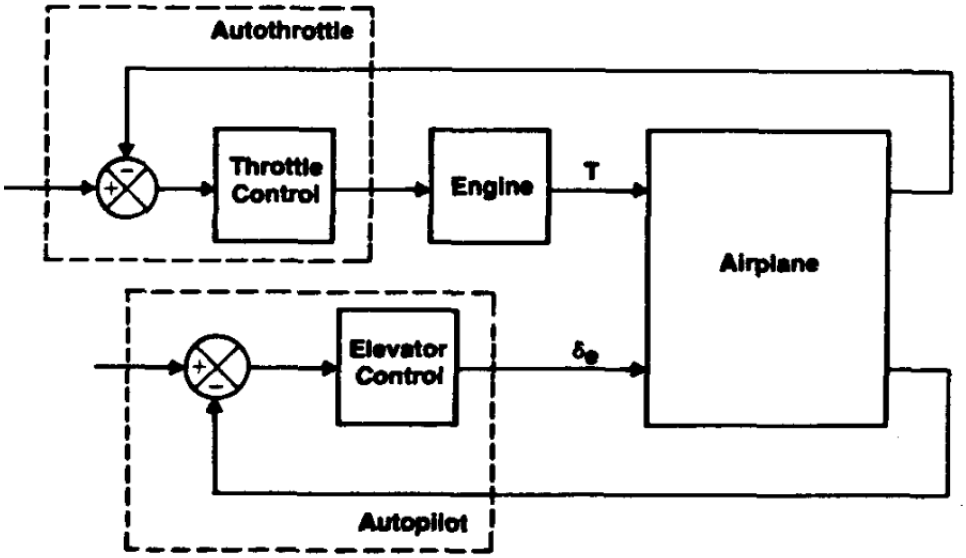


FIGURE 2.3 AFCS Conventionnel [26].

2.2 Les standards de bus avionique physique

Pour faire communiquer les systèmes avioniques, un avion est doté de réseaux contenant différents types de connexion. Une large gamme de standards implémente sa connectivité. Cette section fait un tour d'horizon des bus physiques courants dans un aéronef. Nous présentons les trois types de bus. Ils comportent des similarités et des différences dans leur mesure de sécurité. Nous regarderons les études en termes de cybersécurité qui ont été publiées à leur sujet. Le tableau 2.1 compare l'ARINC 429, le MIL STD-1553 et le bus CAN.

2.2.1 L'ARINC 429

Le standard ARINC 429 est bien la technologie la plus répandue et utilisée aujourd'hui. Il s'agit d'une norme de communication dans l'industrie aéronautique, conçue en 1977, qui assiste à la connectivité des différentes composantes avioniques d'un avion. Elle a été développée par l'Aeronautical Radio Incorporated (ARINC), une association de l'industrie aéronautique américaine, pour standardiser les communications entre les équipements de navigation, de contrôle de vol et de systèmes de gestion de bord d'un avion [3].

Cette norme s'occupe de la connectivité de l'avion. Elle définit les protocoles de transmission de données, ses codes d'identification et ses spécifications électriques pour les connexions physiques et les signaux. Le bus de donnée défini est unidirectionnel (simplex). Les paires de câbles torsadés blindées sont utilisées pour faire communiquer directement des entités de communications. Une de ces entités porte le nom de Line-Replaceable Unit (LRU). On dit de l'architecture de l'ARINC 429, qu'elle est fédérée puisque les LRUs sont interconnectés tout en étant segmentés [3]. À l'opposé des standards de mise en réseau d'aujourd'hui, notamment l'AFDX, un récepteur LRU ne peut envoyer de messages lorsqu'il est en train d'écouter sur un bus simplex. Pour permettre un échange bidirectionnel, il faudra créer une seconde communication simplex où le récepteur et l'émetteur échangent leur rôle [2]. La transmission peut s'opérer à basse vitesse, avec un débit maximal de 12.5 kbit/s ou à haute vitesse à un débit de 100 kbit/s. Une transmission ne peut contenir qu'un seul transmetteur qui peut transmettre à au plus 20 récepteurs.

Un *mot*, dans la terminologie de l'ARINC 429, est un paquet de données codées sous 32 bits. Sa faible taille permet une efficacité de transmission adaptée avec le débit. Le mot est généralement sous format Binary (BNR), Binary Coded Decimal (BCD) ou Discrete. La plupart des messages ARINC ne contiennent qu'un seul mot.

Un mot ARINC 429 est composé de 5 champs [27] :

`Label :` `8 bits`
Les bits 1-8 contiennent les identifiants (ou étiquettes) d'information. Le label identifie le type de données(BNR, BCD, Discrete, etc) et contient les instructions d'interprétation des données

`SDI :` `optionnel 2 bits`
Les bits 9-10 optionnels assignés au Source Destination Identifier. Identifie la source et la destination d'une transmission

`Donnée :` `19 bits`
Les bits 11-29 contenant les données d'information d'un mot. Assez flexible, le type de données est le plus souvent en BNR, BCD ou discret

`SSM :` `2 bits`
Les bits 31-30 assignés au Sign/Status Matrix field. Indique le type de données transmises

`Parité :` `1 bits`
Le bit le plus significatif, l'ARINC utilise la parité comme vérificateur d'erreurs après la réception

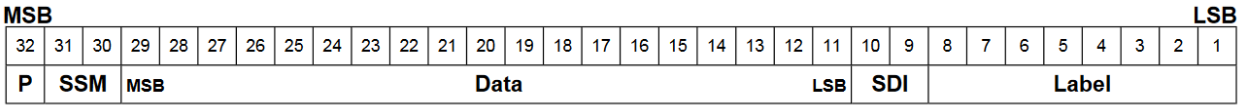


FIGURE 2.4 Format d'un mot Arinc 429 [27].

L'exemple d'un mot de type BNR ARINC 429, sans cahier de charge, présente ci-dessous les informations concrètes que nous pouvons décortiquer. Le paquet ici transmet une altitude de 28, 137 ft depuis l'altimètre d'un avion :

`Label :`	`059`	Altitude
`SDI :`	`00`	Mot de source unique
`Donnée :`	`28 137`	28 137 ft (décimal)
`SSM :`	`00`	Opération normale

Le standard ARINC 429 est fiable, robuste et capable de gérer des données complexes. C'est pour cela qu'on l'utilise dans plusieurs avions commerciaux tels ceux de Bombardier, Embraer, Airbus et Boeing. Cependant, son utilisation est limitée aux systèmes de navigation et de contrôle de vol. Il ne couvre pas les communications de données de bord plus large,

comme les systèmes de divertissement ou de connectivité [3].

La segmentation des fils renforce la sécurité de l'avion en limitant l'échange de données et de commandes électroniques. Toutefois, l'ARINC 429 est une technologie vieille de 1978. Elle a été réalisée sans prendre en compte la cybersécurité et a surtout misé sur la fiabilité et l'efficacité des communications aéronautiques [6].

2.2.2 CAN BUS

Le bus Controller Area Network (CAN) est un protocole de communication automobile et avionique qui comporte plusieurs similarités avec l'ARINC 429. Il est lui aussi un protocole fait pour des réseaux locaux avec fil. Son débit et sa longueur sont similaires au 429 avec un taux de 1 Mbit/s et des mots de 128 bits. Tout comme le 429, CAN a été conçu il y a plus de 30 ans sans prendre en compte la sécurité. Il ne possède pas non plus d'authentification ou de cryptage. Dans le cas d'une différence de valeurs entre un bit transmis et un bit observé par le bus, seul un message d'erreur remonte. Le standard obéit à la règlementation ISO 11898-1 [17]. Enfin, ce bus est à communication bidirectionnelle (*duplex*) et est utilisé pour transmettre des données entre un contrôleur de bus et les *remote terminals* (RT) grâce à sa topologie maître-esclave. Le bus a pour avantage de demander peu de câblage, d'être faible en coût, peu complexe et d'offrir une transmission sur deux lignes. Il est toutefois difficile de le sécuriser puisqu'il n'authentifie pas les paquets [28].
Le bus CAN enrichit notre état de l'art puisque des chercheurs peuvent plus facilement avoir accès à de vraies voitures que de vrais avions pour leur simulation. En effet, à part quelques programmes du département américain, peu d'individus sont prompts à fournir un véritable avion pour des tests de pénétrations [7]. C'est pourquoi la plupart des recherches publiques sont faites avec ce moyen de transport pour un coût d'étude moindre. Le bus CAN a ainsi été plus étudié et une meilleure connaissance technique dans la littérature [23]. Étudier CAN sert dans notre recherche sur l'ARINC 429 puisque sa forte similarité avec l'ARINC 429 nous fournit davantage d'information sur le comportement de ce dernier face aux attaques informatiques.

2.2.3 MIL-STD-1553

MIL-STD-1553 est un autre standard de bus, couramment utilisé dans l'aéronautique militaire, qui a été développé par l'US Air Force. On le retrouve dans les équipements de systèmes de contrôle de vol, de communication ou de détections d'erreurs. Les spécifications des connexions sont bidirectionnelles et multiplexées sous une diffusion *broadcast* ou commande-réponse.

Ses avantages pour le militaire reposent sur son interopérabilité, ses performances, sa fiabilité et sa tolérance aux pannes. Toutefois, son gros défaut est qu'il s'agit d'un réseau centralisé composé d'un point de défaillance unique. Ceci vient potentiellement affecter sa fiabilité puisqu'un contrôleur compromis peut paralyser l'entièreté d'un système de bus, d'où la nécessité de redondance sur l'architecture [29]. La dernière mise à jour du MIL date de 2018 avec MIL-STD-1553C. Ce standard n'a toutefois pas été conçu pour prendre initialement la cybersécurité comme critère, n'est doté d'aucun cryptage et enfin est incapable de tracer un message [30].

	ARINC 429	**MIL STD-1553**	**Bus CAN**
Basse Vitesse	12.5 kbps	1 Mbps	125 kbps
Haute Vitesse	100 kbps	1 Mbps	1 Mbps
Câble	Unidirectionnel torsadé	Bi-directionnel torsadé	Bi-directionnel torsadé
Nombre d'unités	Max de 1T20R	Jusqu'à 32 RTs	Max de 32TyR
Dernière révision	Plus de 30 ans	2018	Plus de 30 ans
Nombre de bits	32	128	20
Critère de Sécurité	NON	NON	NON
Architecture	Fédérée	Partiellement IMA	ISO 11898

Tableau 2.1 Résumé des caractéristiques des bus avioniques.

2.2.4 L'Avionique Modulaire Intégrée (IMA) et l'AFDX

L'Avionics Full-Duplex Switched Ethernet (AFDX) est un standard de bus pour les systèmes de bord dans l'industrie de l'aéronautique destiné à remplacer les vieux systèmes de bus des avions mentionnés précédemment. Il est basé sur le standard de réseau Ethernet sur lequel les réseaux locaux (LAN) sont formés. La transmission est faite en temps réel et son architecture en étoile garantit la connexion des différentes composantes avioniques de l'avion. L'augmentation de la connectivité sans-fil risque de rendre l'AFDX plus vulnérable à des attaques informatiques [25]. La nature du bus unidirectionnel ne peut tout de même pas être garantie. Les mesures de mitigations doivent s'adresser à toutes les sources de données ou d'interférence sur les bus.

On assiste ces dernières années à la tendance de vouloir remplacer l'ARINC 429 par des communications Ethernet et AFDX. En consolidant les logiciels en IMA, on obtient un meilleur échange de données entre l'avion et le sol. Cela facilite surtout la maintenance, la navigation ainsi que les opérations. Il comble aussi au besoin de réduction du poids et de l'infrastructure réseau. Plus important, l'IMA définit une couche d'abstraction entre le logiciel et le matériel [31]. L'AC 20-156 «Aviation Data Bus Assurance» est une Circulaire d'information (CI) faite pour guider le remplacement des connexions des bus unidirectionnels par des bus bidirectionnels. Ce guide pratique est utile lors de changements majeurs dans l'installation d'un aéronef avec des technologies de bus de données plus complexes et hautement intégrés. On s'en sert par exemple pour réaliser l'interconnexion entre l'ARINC 429 et l'AFDX [25].

2.2.5 L'ARINC 429 comme pierre angulaire vers la sécurisation des bus

L'aviation vient tout juste d'amorcer un parcours de mitigation des risques de cyberattaques sur les bus. Encore beaucoup d'interrogations sont présentes quant au plan à entreprendre. Nous considérons qu'évaluer l'ARINC 429 est la meilleure optique actuelle qui pourra par la suite évoluer vers des bus plus sophistiqués. Malgré l'arrivée de nouveaux protocoles mieux adaptés aux avions modernes, comme l'AFDX, il est peu probable que l'ARINC 429 disparaisse subitement. Sa fiabilité a été mainte fois démontrée. Ce standard demeure en effet pertinent pour des scénarios qui font intervenir de simples signaux et qui demandent une excellente latence. Même des avions modernes, tel l'Airbus A380, l'utilisent encore [31].

La faible taille des messages minimise le délai durant le processus d'envoi et favorise la synchronisation puisqu'il ne peut y avoir de file d'attente ou de réordonnancement. Il demeure donc aussi un réseau de secours pour l'AFDX. Enfin, aucun autre standard de communication dans l'avionique n'offre autant de réactivité aux tâches à la microseconde près. Somme toute,

il est pertinent de maintenir une sensibilité à la sécurité pour protéger les communications ARINC 429 contre les menaces potentielles. Pour notre EEA, nous décidons d'appliquer ainsi les techniques d'ingénierie des connaissances adéquates pour ce standard de bus.

2.3 L'ingénierie des connaissances

L'ingénierie des connaissances est une discipline qui s'intéresse à représenter la connaissance dans les systèmes informatiques par la définition, la spécification et de la description des éléments d'information, de leur réseau et de la structure de données [12]. Comme approche on retrouve notamment l'intelligence artificielle (IA) et la logique formelle. Nous penchons toutefois ce mémoire vers la logique descriptive, qui est utilisée pour représenter des domaines par le biais d'ontologies.

2.3.1 Les modèles de données sémantiques

Les premiers modèles de données sémantiques ont été initialement introduits par Peter Chen en 1976. On représente la connaissance en une forme plus expressive qu'un modèle conceptuel [12]. Les modèles de données sémantiques trouvent leur place dans la science des données et de l'ingénierie logicielle. Ces modèles de connaissance représentent des collections de **concepts** abstraits. Les concepts sont reliés par des relations qui consistent en un ensemble de **règles**. Un concept va aussi souvent posséder des attributs comme des ensembles de valeurs ou des définitions. Les règles, quant à elles, vont être des contraintes spécifiant, entre autres, le nombre de connexions d'un concept à un autre [32].

Le Web sémantique permet d'expliquer, d'échanger et d'intégrer l'information du Web à des machines, tels des automates ou des algorithmes [33]. Grâce au modèle de données RDF, standardisé par le World Wide Web Consortium (W3C), les données du Web sémantique décrivent les **concepts** et les **règles** sous-jacentes aux pages HTML. Elles spécifient aussi les informations implicites contenues dans les médias ou des bases de données grâce à des représentations de connaissances publiquement accessibles [32]. On représente la connaissance à travers RDF en décrivant des ressources par le biais d'un graphe. RDF repose sur le concept de triplet *sujet-prédicat-objet* (p [s, o]) qui forme le graphe de données. En d'autres mots, un sujet s possède un objet o qui a pour un prédicat un attribut p [32]. L'exemple ci-dessous présente le triplet d'un avion qui vole près d'un aéroport. Comme on le voit, la sémantique ressemble à celle d'une phrase dans la langue française :

— Sujet : Avion (Objet)
— Prédicat : "vole près de" (Action)

— Objet : Aéroport (Endroit)

2.3.2 Modélisation par le biais d'une ontologie

L'ingénierie des connaissances aura recours à une ontologie pour réaliser la connexion entre des concepts de domaines variés. L'ontologie répond en effet au besoin définir de manière explicite et structurée le vocabulaire d'un domaine de connaissance.

La notion de l'ontologie utilisé ici sort du cadre de celle utilisée en philosophie, qui, elle, traite plutôt de l'étude métaphysique de la nature de l'être et de l'existence. Dans les sciences de l'informatique, on parle plutôt d'une représentation formelle qui décrit des entités et des relations d'un domaine de connaissance et qui sert de pont de communication entre l'humain et la machine. Une bonne ontologie supporte l'interopérabilité, la coordination des mandats et les standards de son domaine. À la base, elle a été développée comme un formalisme pour la représentation des connaissances en IA afin de représenter des informations stockées dans de grandes bases de connaissances [34].

Les éléments principaux d'une ontologie sont les *classes* pour catégoriser les différents types d'entités du domaine, les *propriétés* pour la notion de relation et les individus qui sont les *instances* de classes. S'ajoutent aussi les axiomes terminologiques qui énoncent les propriétés des concepts et des règles en plus de leurs relations [32]. S'en suivent aussi les assertions de faits. Une classe représente un ensemble d'entités, de sujets ou de ressources qui ont des propriétés que tous partagent [32]. Une propriété définit les règles et les rôles entre des classes, des instances. Une propriété peut aussi exister entre une classe et une valeur ou d'une instance avec une valeur. En d'autres mots, il existe deux types de propriétés : les propriétés d'objets , qui relient une entité à une instance d'une autre classe (*object type property*) et les propriétés de type de données (*data type property*), qui attribuent à une entité une valeur (numérique, chaîne de caractères, etc.) [11].

Une ontologie peut être représentée en RDF qui offre la description des relations intrinsèque des triplets p[s, o] eux-mêmes [35]. La construction d'une classe va suivre la définition formelle suivante :

— Classe : Ensemble, collection ou concept. Exemples : Avion, Pilote, Aéroport
— Propriété : Aspect, caractéristique, propriété ou paramètre d'un objet ou d'une classe. Exemples : «destination», «vol près de», «est piloté par»
— Individus : Instance ou objet de base. Exemples : Boeing 747, Pilote Hans Obas, YUL

2.3.3 La notion de TBOX et de ABOX

La représentation de la connaissance peut être appliqué dans une base de connaissance afin de construire des déclarations en logique descriptive. Elle se subdivise en deux principaux composants qui sont la terminologie [TBOX] et l'assertion [ABOX].

D'une part, la TBOX définit les classes, les propriétés d'un domaine en plus de spécifier leurs relations entre elles. Dans les paramètres d'une base de connaissances, elle correspond aussi à la couche *schéma*. Il s'agit ainsi du côté «statique» de l'ontologie. Un axiome terminologique peut par exemple contraindre le domaine [*rdfs : domain*] ou le codomaine [*rdfs : range*] d'une classe. Il peut également introduire un concept pour une description complexe. Dans cet exemple, le concept de CommercialAircraft définit un avion piloté à des fins commerciales :

$$CommercialAircraft \equiv Aircraft \sqcap (\forall hasBusStandard.(ARINC429 \sqcap AFDX)).$$

L'axiome définit le concept par l'union de deux restrictions : la première indique que l'entité est une instance de la classe Aircraft, et la seconde indique qu'elle ne peut être liée, par la propriété hasBusStandard, qu'à une instance de ARINC429 ou de AFDX.

Une TBOX plus expressive peut aussi faire intervenir une déclaration d'axiome plus générique tel que :

$$\exists hasAvionicsBusStandard.AvionicsBusStandard \subseteq Aircraft\ ,$$

qui déclare que seuls des avions comportent des standards de bus avioniques. Cette manière de définir la TBOX la rend plus lisible [32].

D'autre part, la ABOX consiste en un ensemble fini d'assertions pour la classification d'instances et de leurs propriétés [32]. Elle correspond au côté «dynamique» de l'ontologie et aux données d'une base de connaissance. Par exemple, l'assertion :

$$CommercialAircraft[BOEING747],$$
$$ARINC429[BOEINGBUS_A],$$
$$hasBusStandard[BOEING747, BOEINGBUS_A].$$

indique, en logique descriptive, qu'un avion Boeing 747 appartient au concept de *CommercialAircraft* et qu'il intègre le bus Boeing A qui lui-même est une instance du standard

ARINC429.

Combinées, la TBOX et la ABOX permettent d'inférer, grâce à un raisonneur adéquat, de nouvelles connaissances. Ce dernier navigue à travers les classes, les instances et leurs propriétés en produisant des résultats aux questions posées à l'ontologie qui prendront la forme de requêtes [12].

La figure 2.5 présente un exemple d'ontologie, sous forme de représentation graphique, combinant TBOX et ABOX. Dans la TBOX, on représente un avion commercial comme étant un avion qui a recourt au standard de bus AFDX ou ARINC 429. On y définit en plus l'avion comme une sous-classe d'un système d'ingénierie. Le Boeing 747 est enfin instancié dans la ABOX comme un individu (*individual*) de la classe de l'avion commercial.

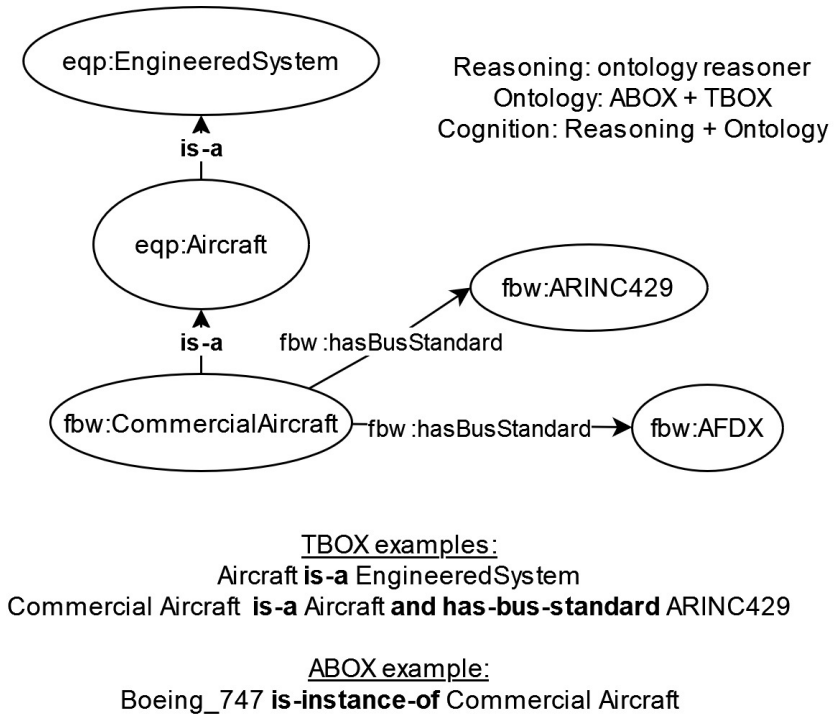


FIGURE 2.5 Relations, classes et rôles d'une ontologie sur les types d'avions [12].

2.3.4 OWL

Construit au-dessus de RDF sur la couche logique, le Langage Ontologique de Web [OWL] a engendré des ontologies plus détaillées grâce à une sémantique formelle. Ses capacités à exploiter des résultats en logique formelles offrent des inférences plus riches [14,32]. OWL est aussi apte à utiliser des raisonneurs et offre plus de fonctions que RDF en étant tout autant standardisé et interopérable. Il est à ce jour le principal langage ontologique utilisé [36]. Une

ontologie OWL consiste ainsi en un ensemble d'axiomes capable d'inférer des relations de subsomption ou d'équivalence. De plus, OWL offre de définir plus précisément des propriétés. Il est en effet possible d'affirmer qu'une propriété est transitive, fonctionnelle, inversement fonctionnelle, symétrique ou asymétrique.

2.3.5 Le domaine de l'aviation

L'aviation est en soi un domaine de connaissance. Si on prend le cas d'un aéroport, il s'agit d'une infrastructure comprenant plusieurs sous-systèmes, telle la gestion des passagers, des bagages ou des services de l'équipe de vol. Ces sous-systèmes s'échangent de l'information. L'objectif d'interopérabilité est donc déterminant.

NextGen et SESAR ont proposé l'idée qu'intégrer une ontologie unifiée est bénéfique pour l'aviation [21]. Ceux-ci avancent quelques sous-domaines d'intérêts. Pour commencer, on a la coordination des capteurs physiques, entre autres ceux du positionnement ou de la navigation, qui offrent de meilleures performances. Ensuite, les ontologies peuvent aussi servir dans le sous-domaine des communications pour les signaux d'appel ou le Notices to Airman (NOTAM). Enfin, les technologies en Web sémantique peuvent améliorer les rapports de situations, lors de défaillances, de cyberattaques ou de conditions météorologiques [11]. Notre recherche poursuit ces postulats en intégrant l'avionique, ontologie et cybersécurité.

2.4 La cybersécurité

Nous terminons ce chapitre en offrant une définition des objectifs et des attaques de sécurité les plus courantes visant les systèmes cyberphysique afin de mieux comprendre les concepts que nous aborderons par la suite.

2.4.1 Les couches OSI

Le modèle OSI est un standard définissant les sept couches d'application s'exécutant à travers des composantes de réseaux et communiquant entre elles. Les trois couches réseau principales des systèmes d'aviation cyberphysique sont la couche application, la couche de transmission et la couche physique.

Tout d'abord, la couche application organise, analyse et ajuste le matériel informatique physique pour atteindre un comportement désiré. Dans le domaine de l'aviation, cette couche contient des données sensibles à propos des compagnies aériennes, des passagers et des aéroports. Une attaque dans la couche application engendre d'énormes impacts. Ensuite, on

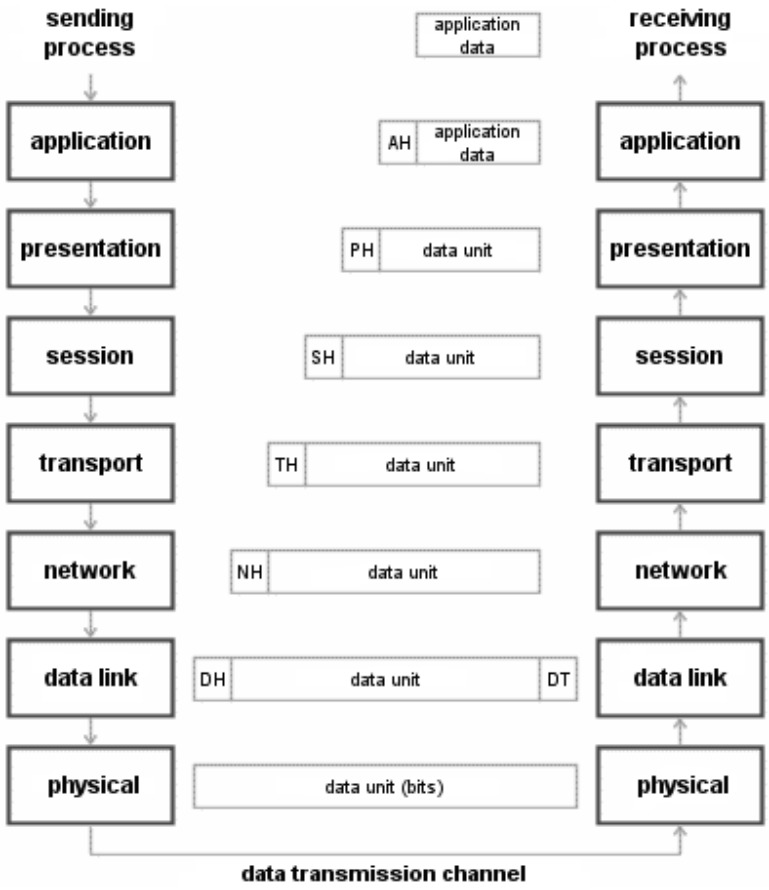


FIGURE 2.6 Représentation des 7 couches du modèle OSI [37].

retrouve la couche de transmission qui est située entre la couche application et physique. Les données de cette couche sont de petits paquets transmis et transférés. Les communications sans-fils sont affectées par les attaques visant cette couche. Enfin, la couche physique récupère les informations des capteurs et caméras. Il agit sur le monde physique afin d'obtenir le comportement désiré. Les données sont envoyées à la couche application qui décidera des actions à exécuter par l'actuateur [38].

2.4.2 L'objectif de cybersécurité

Les trois critères assurant la sécurité sont la confidentialité, l'intégrité et la disponibilité. La confidentialité assure la vie privée des utilisateurs et protège leurs données contre entre autres l'usurpation. L'intégrité assure que les données sont envoyées au destinataire désiré et sans être modifié par un élément malicieux. La disponibilité s'assure que seuls les utilisateurs valides ont accès aux ressources de réseau. Un système informatique est défait et vulnérable quand les critères de confidentialité, d'intégrité et de disponibilité sont compromis [30].

2.4.3 Définition des attaques de cybersécurité les plus courantes

Usurpation (*Spoofing*) : L'attaque tente de tromper le système, telle l'identité d'un individu ou d'un paquet. En d'autres mots, on modifie les données qui semblent venir de source valide [39]. Des exemples d'usurpation en aviation sont la création d'avions fantômes ou les viols de lois physiques [40]. Un adversaire fabrique des messages usurpés pour interférer avec les opérations de l'avion et alors affecter la sécurité du vol. Le défi d'un attaquant pour un bus est de déterminer le temps d'inactivité pour transmettre ses paquets. La confidentialité, l'intégrité et la disponibilité peuvent être affectées [15].

Logiciel malveillant : Ces logiciels sont conçus pour causer des dommages au système et compromettre sa sécurité. Les vers, virus et cheval de Troie en sont des exemples [39].

Déni de service (DOS) : Un DOS manipule les messages ou les comportements d'un système (collision, échec, débordement de tampon, etc.). La forme la plus courante de DOS consiste à inonder avec énormément de trafic les zones à risque qui s'assurent de la sureté de l'avionique. Elle attaque la disponibilité en épuisant la bande passante, en brouillant les communications ou en dérangeant l'opération de réseau [41]. La disponibilité est souvent affectée. Le brouillage (*jamming*) est une sous-catégorie de DOS où l'on diminue le pourcentage de communication réussi en dessous de 30 % en intégrant du bruit. [30]. Une autre sous-catégorie est l'inondation de paquet (*flooding*) qui consiste à sursaturer le réseau d'un énorme trafic.

Contrefaçon (*Physical Tampering*) :Accéder aux composantes déployées dans le système et modifier leur circuit en insérant par exemple du contenu malicieux dans une partie de l'avion, un logiciel ou une base de données [3, 42]. Son but serait par exemple d'affecter un avion dans le but de transmettre de fausses informations à la tour de contrôle.

Attaque sur la chaîne d'approvisionnement : Un avion moderne contient plusieurs millions de composantes provenant de fournisseurs, de sous-fournisseurs et de sous-fournisseurs partout dans le globe. Des logiciels malicieux peuvent être introduits dans n'importe quel système avionique par un acteur fourbe [43]. Seulement en 2010, la US Navy a découvert avoir acheté 59,000 micropuces contrefaites [44].

Attaque par rejeu : L'adversaire écoute clandestinement une communication et renvoie à un récepteur légitime. L'ARINC 429 est protégé contre ce type d'attaque [6].

Écoute clandestine (*Eavesdropping*) : Ce type d'attaque intercepte les données transmises sur le réseau. Avec l'absence de cryptage, il devient facile d'écouter sur une même fréquence et de capturer des flux de données provenant d'un avion. Cette attaque ne cause que très peu de dommage à un appareil [38]. Elle peut quand même obtenir des données

précieuses comme des avantages commerciaux ou des données de sécurité sur ces cibles pour ensuite mieux exécuter des attaques.

CHAPITRE 3 REVUE DE LITTÉRATURE

Ce chapitre survole les recherches jugées pertinentes afin d'avoir une vision globale de la sécurité du bus ARINC 429 et de l'ontologie.

3.1 Sécurité de l'ARINC 429

L'ARINC 429, n'a pas, comme on le rappelle, pris en compte la cybersécurité comme objectif dans sa conception. Cette section présente l'état de l'art des découvertes récentes liées à la sécurité autour des bus avioniques.

3.1.1 Bancs de test

Lorsqu'une faute d'avionique se produit, elle peut causer un grave accident durant le vol. Un analyseur de bus empêche de tels scénarios de se produire durant la phase de maintenance. L'étude de [29] présente un prototype de test du bus ARINC 429. Il est apte à mesurer les caractéristiques électriques pour mieux surveiller les performances de l'état de fonctionnement. Malgré tout, il s'agit d'un prototype adapté pour la maintenance qui n'est pas viable dans un vol nécessitant une analyse en temps réel. En 2013, lors de la conférence «Hack in the Box» à Amsterdam, Hugo Teso [45] démontre qu'il est possible de contrôler un avion avec de l'équipement disponible sur eBay. Son attaque exploite les communications de l'ADS-B, un système de contrôle du trafic aérien, et l'ACARS pour attaquer le FMS. Il réussit non seulement à écouter des messages, mais aussi à les manipuler et à envoyer du code malicieux. Ses manipulations incluent la modification du plan de vol, des prévisions météo et d'identifiants sur l'avion. Pour prévenir les vulnérabilités des cybermenaces liées aux réseaux sans-fils des systèmes cyberphysiques, Kumar et al. [46] présente un *framework* qui analyse le chargement de données avioniques et utilise des tests de pénétration avec les outils Metasploit Pro et BackTrack. L'étude conclut qu'il est impératif de recourir à des évaluations de risque et de développer des plans de mitigations.

Le banc d'essai Triton [47] se démarque en instanciant de multiples sous-systèmes de l'ARINC 429. Son objectif y est d'observer les comportements agrégés des systèmes d'aviation et d'identifier leurs impacts potentiels envers la sureté des vols. Son atout repose sur sa capacité à virtualiser l'ARINC 429 et rendre la simulation plus fidèle. Elle cible comme vecteurs d'attaques l'ACARS et le chargement de donnée des Loadable Software Airplane Parts (LSAP), une sous-catégorie de FLS. D'une part, on juge intéressant l'ACARS pour évaluer si des

entrées malicieuses depuis une radio VHF ne pourraient pas influencer les opérations du système de contrôle de vol. L'équipe simule alors un message malicieux depuis l'unité de Gestion des Commandes (CMU). D'autre part, le chargement de données aéroportées (LSPs) est un autre vecteur puisqu'il peut directement mettre à jour le logiciel d'un LRU et ses données d'assistance. Les LRUs, eux, sont incapables de détecter une mise à jour trafiquée par rapport à une mise à jour vulnérable. Ce qui rend la situation inquiétante, c'est que les LRUs ne sont parfois pas capables de détecter une mise à jour corrompue ou des vulnérabilités depuis l'Aeronotical Data Link (ADL). Une attaque peut avoir lieu de ce fait quand l'avion est au sol. Pour finir, l'intégration de plusieurs LRUs sur Triton impose de minces contraintes de délais. Certaines capture des caractéristiques électriques et horaires des signaux sont ainsi manquées.

Yahalom et al. [43] présente un jeu de données d'un trafic simulé de paquets du bus MIL-STD-1553 avec des messages normaux et des messages malicieux injectés dans un bus par un terminal. Même s'il ne s'agit que de simples données, ce jeu *open source* contribue à aider dans la détection de cyberattaques. En effet, il permet d'évaluer et de comparer des IDS avec ses séquences de données.

Les petits avions personnels sont rarement stationnés dans des zones surveillées. Ils sont plutôt mis dans des garages ou des stationnements ouverts ou publics. Patrick Kiley [28] démontre sur un banc de test d'un bus CAN (voir figure 3.1) que l'accès physique d'un avion permet à un individu d'envoyer de fausses données ou manipuler le voltage, ce qui peut engendrer un atterrissage d'urgence ou une perte de contrôle catastrophique. On y produit une attaque par usurpation de paquets sur l'Attitude and Heading Reference System (AHRS), une attaque par rejeu sur le PFD et un désengagement nuisible de l'auto-pilote. [28]. Patrick conclut que la meilleure solution serait d'intégrer une passerelle CAN.

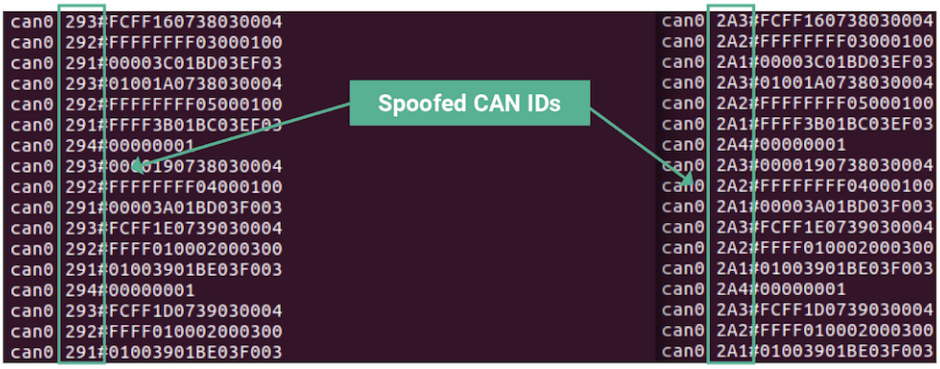


FIGURE 3.1 *Spoofing* du bus CAN d'un AHRS [28].

Deux échantillons d'applications sont produits par True et al. [48] pour une évaluation qualitative de l'architecture d'un avion en termes de cybersécurité. Les environnements de simulations choisis sont un cas de négociation de trajectoire ciel à terre et un cas d'instruction taxi. Des composantes avioniques prises en compte sont l'EFB, les interfaces Aircraft Interface Devices (AID) et la couche de liaison de données du réseau IP. Toutes ces composantes communiquent à travers l'Airline Information Services Domain (AISD). L'EFB représente une porte d'entrée pour deux menaces. On retrouve le déni de service qui est surtout causé par une protection inadéquate des connexions réseaux de l'AID et l'attaque par homme du milieu qui est sévère sur l'EFB lorsqu'on fait intervenir un logiciel malicieux dans le réseau. Des mesures de sécurité sont proposées. Pour les dénis de service, on peut filtrer les paquets ou limiter le débit. Pour ce qui est des programmes malveillants sur l'EFB, on peut se fier aux standards de sécurité de l'industrie informatique avec des démarrages sécuritaires, des protections logiciels pendant l'exécution, du chiffrement ou un contrôle de l'accès du noyau. En parallèle, Alex Lomas produit une simulation d'attaque sur l'EFB avec des données trafiquées pour persuader le FMS que l'avion est beaucoup plus léger qu'il ne l'est vraiment [49]. Par conséquent, les *V-speeds* recommandés sont mis beaucoup trop bas. L'action est évaluée comme étant apte à produire un toucher de queue (*tailstrike*). Ceci peut causer un incident, voire un accident.

De Santo et al. [30], produisent un logiciel qui analyse le bus MIL-STD 1553. Une vulnérabilité viable repose sur la division du temps lors du multiplexage temporel délégué au contrôleur. L'équipe de recherche conclut qu'une attaque par usurpation de paquet sans créer de collision avec le trafic, en instaurant une communication aux bons délais avec un composant voyou. À l'inverse, causer des collisions et produire un brouillage des communications afin de rendre impraticable l'échange de données est aussi faisable selon l'implémentation sur les couches logiques et de la gestion de trafic réseau.

Strohmeier et al. [7] conçoivent un banc de test de l'ARINC 429 certifié avec du véritable matériel informatique d'avion plutôt qu'avec des composantes simulées ou émulées. On y évalue les cyberattaques connues sur l'ADS-B et le GPS qui visent les communications de l'avion. L'équipe recommande aux chercheurs de se concentrer sur les attaques visant des protocoles standards indépendants des manufacturiers. Une difficulté mentionnée est que la plupart des fournisseurs évitent de vouloir stresser la sécurité de leur équipement et préfèrent se tourner vers l'obfuscation. Seul le manufacturier Garmin a été ouvert à collaborer avec les chercheurs.

3.1.2 Sommaire : Une avionique vulnérable

Somme toute, les bancs de tests montrent que l'avionique comporte diverses vulnérabilités qui peuvent se répercuter sur les bus. Le FMS, l'EFB, l'ACARS et les FLSs sont entre autres des vecteurs d'attaques vulnérables à l'usurpation d'identité, le déni de service et l'attaque homme du milieu. La sécurité des couches logiques du modèle OSI doit aussi particulièrement être surveillée contre des attaques sur la chaîne d'approvisionnement, en plus des attaques ciblant les protocoles standards administrés sur l'avionique de plusieurs modèles d'aéronefs.

En tant que chercheur en sécurité de l'aviation, il serait fragile de se reposer uniquement sur une seule dimension de la sécurité pour se protéger. Même si ces attaques ont peu de risque de faire écraser un avion, le non respect de la «défense en profondeur» principe-clé en cybersécurité, compromet la sécurité d'un vol.

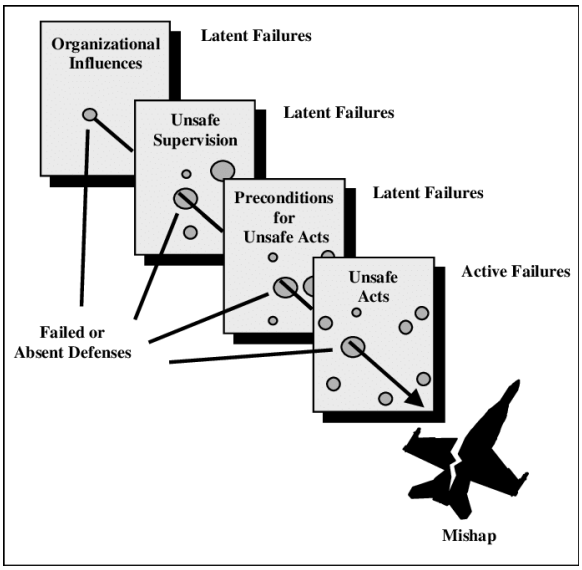


FIGURE 3.2 Le modèle du fromage suisse de J. Reason des causes d'erreurs humaines [50].

Les simulations d'attaques informatiques durant un vol [49] ont montré qu'une simple erreur d'entrée d'un pilote sur l'EFB est apte à causer de graves accidents. Une combinaison d'événements consécutifs s'est produite pour qu'un écrasement d'avion survienne. On appelle ce principe, celui du fromage suisse de J. Reason [50]. Chaque facteur est aligné à un autre et lorsque les trous de nos tranches de gruyères s'alignent, les conditions sont établies pour causer un événement catastrophique.

La même situation se produit dans le cas de la cyberdéfense. Que ce soit donc à cause d'une augmentation soudaine de la charge cognitive d'un pilote, d'un hangar privé mal surveillé, d'un ouvrier de maintenance malicieux, de revendications de protocoles de sûreté [7], de

redondance ou de COTS avec une défense boîte noire provenant d'un fournisseur qui n'est géré par aucune agence gouvernementale, toutes ces mesures ne sont pas optimales ou adaptées pour consolider la sécurité de l'avion. Nous clamons même qu'il ne s'agit pas tout simplement de mesures de sécurité.

Un moment d'inattention clé suffit pour que chaque couche de protection actuelle soit franchie. La faisabilité de ces attaques soulève ainsi une nécessité à mesurer la mitigation face aux accès physiques à bord de l'aéronef. La prochaine section s'intéresse aux contre-mesures développées pour protéger les bus de communication ARINC 429, CAN et MIL-STD-1553.

3.2 Des contre-mesures pour faire face aux cyberattaques sur les bus

Certains chercheurs se sont positionnés pour sécuriser les bus avioniques. Nous avons catégorisé les solutions existantes en sous-sections. Certaines d'entre elles combinent parfois plus d'une catégorie.

3.2.1 Le chiffrement

Eidsness [51] propose dans un livre blanc d'avoir recours à la signature cryptographique des capteurs ARINC 429 dans le but d'authentifier un transmetteur susceptible de produire de l'usurpation de paquets ou du Déni de service (DOS). L'ARINC 429 possède une haute impédance quand le transmetteur est inactif, ce qui le rend vulnérable à l'ajout physique d'un second transmetteur actif capable de réaliser lesdites attaques. Corrompre la parité avec un second transmetteur malicieux attaché au bus vient affecter sa disponibilité. La contre-mesure proposée, la signature cryptographique, est rétrocompatible avec les anciennes interfaces de l'ARINC 429. Pour diminuer la puissance de calcul causée par chiffrement et être à temps réel, il propose de chiffrer la somme de contrôle *checksum* des paquets plutôt que les paquets eux-mêmes. Malgré une augmentation de la sécurité des données, le chiffrement demeure quand même critiqué puisque l'implémenter sur l'ARINC 429 implique des modifications ardues sur son architecture en plus d'un coût de la puissance de calcul [17].

3.2.2 Empreinte matérielle (*Hardware Fingerprinting*)

Markevich et al. [6] utilise l'empreinte matérielle de l'avionique pour détecter si son intégrité a été compromise. Combiné aux chaînes de Markov, cette solution diminue le taux de faux positifs en évaluant le nombre de fausses alarmes par seconde. Il identifie les différents transmetteurs et receveurs même s'ils proviennent du même manufacturier. Il est de plus efficace contre les attaques techniciennes où un LRU légitime est échangé par une unité voyou. L'ex-

périence corrobore aussi que l'ajout d'un deuxième récepteur affecte moins le signal que d'en remplacer un. Comme l'empreinte matérielle cible la couche physique et de transport, elle ne prend pas en compte la détection basée sur l'analyse du contenu d'un paquet ARINC 429.

Une autre solution [34] capture le signal électrique du bus MIL-STD-1553B en ajoutant un détecteur d'anomalies léger pour surveiller les messages transmis sur la couche application. La solution applique des algorithmes d'apprentissage machine. Le problème toutefois repose sur les attaques non détectées qui utilisent les données ou le statut des mots pour usurper des composantes. On ne peut pas non plus physiquement authentifier un composant.

3.2.3 Les systèmes de détection d'intrusion (IDS)

L'IDS est un logiciel automatisé et désigné pour surveiller ou détecter l'utilisation de ressources malicieuses ou non autorisées. [9]. Un IDS est souvent utilisée pour réajuster la sécurité d'un système n'ayant préalablement pas pris en compte le critère de sécurité.

Il peut être d'une part basé sur l'hôte (HIDS). Les données y sont collectées par un système informatique individuel, soit un hôte, sans rentrer par le réseau. Dans notre cas, les hôtes prennent la forme de composantes avioniques. Les opérations d'un système à l'intérieur de l'hôte sont ainsi surveillées, incluant celle du système d'exploitation, des applications de transactions, des entrées-sorties et même du trafic réseau. L'IDS peut d'autre part être basée sur le réseau (NIDS). On collecte cette fois passivement des données d'un trafic à des points stratégiques comme sur les passerelles ou les commutateurs. On permet alors la détection d'équipements hérités et d'autres systèmes informatiques qui n'aurait pas été possible avec un HIDS [9].

Dans leur papier, Ryon et al. [9] apportent un prototype d'IDS réseau sur les bus basé sur les exigences. Pour évaluer sa faisabilité, des paquets ARINC 429 et AFDX sont capturés dans un réseau. Ils établissent dans leurs expériences que leur prototype est capable d'y détecter un trafic malicieux en prenant comme critère d'efficacité le taux de faux positifs. Les performances sont toutefois diminuées lorsqu'on augmente le nombre de paquets malicieux, ce qui peut laisser ce prototype vulnérable aux attaques par déni de service. Les auteurs avancent qu'une solution serait de diminuer le taux de traitement des paquets. Pour terminer, leur synthèse exprime deux zones d'intérêts à explorer pour de futures recherches sur l'IDS. Une première zone serait celle de l'interaction entre l'humain et la machine. Même si un pilote ne doit pas obligatoirement avoir une compréhension forte des IDS, il doit pour autant bénéficier de son mécanisme pour l'aider à prendre une décision intelligente lors d'une attaque. Le processus de notification du pilote, sa réaction et la quantité d'automatisation de l'appareil sont toutes des questions viables qu'on invite les parties prenantes à se poser. La deuxième zone

d'intérêt porte sur la génération des exigences en cybersécurité de l'avionique. Des approches standardisées par la recherche pourront réduire les dissonances en termes de lisibilité qu'une interface pose entre l'humain et la machine.

Bozdal et al. [17] examinent à travers leur revue de littérature des solutions basées sur l'entropie pour se prévenir face aux cyberattaques sur le bus CAN dans le réseau routier. Une autre solution présente analyse les intervalles de temps entre les messages, tandis qu'une autre analyse les empreintes de décalage de temps des horloges. On y conclut que l'IDS est la solution la plus prometteuse pour cibler les attaques sur le bus puisqu'elle ne modifie pas le contrôleur et n'augmente pas le trafic réseau contrairement à la segmentation du réseau (que les bus simplex de l'ARINC 429 appliquent intrinsèquement), au chiffrement ou à l'authentification. Même si elle n'offre pas une sécurité complète, elle protège de la plupart des scénarios, le tout avec une faible puissance de calcul. Un autre IDS [52] analyse cette fois le trafic pour déterminer si les messages passés sont malicieux et s'ils correspondent à des attaques connues d'usurpation, de DOS ou de vol d'identité. Cette étude obtient de bons résultats en termes de précision. La logique des paquets n'est toutefois pas abordée.

On remarque que des IDS plus modernes ont souvent recours à l'IA. Stan et al. [42] se positionne sur la création d'une contre-mesure et propose un IDS basé sur l'apprentissage machine du signal qui analyse le temps d'intervalle et la fréquence des messages MIL-STD-1553 du jeu de données de Yahalom et al. [43]. Cette approche permet ainsi de détecter les cyberattaques sur le bus 1553 avec un taux de 98 % sans changer la topologie du bus. On retrouve comme attaque le DOS, l'usurpation, et la manipulation du temps d'arrivée des paquets. Il ne couvre toutefois pas les attaques exploitant la logique des données elles-mêmes, une attaque sur la chaîne d'approvisionnement ou le statut des mots. De plus, cette solution ne peut pas physiquement authentifier un équipement.

Levy et al. [15] propose un système de protection contre les attaques informatiques, AnoMili, qui s'inspire du principe de défense en profondeur et utilise les données de Yahalom et al. [43]. Ce système est composé de quatre sous-systèmes : un IDS, un mécanisme d'empreinte matérielle, un détecteur d'anomalie et un engin d'explication d'anomalie. Tout d'abord, l'IDS détecte les dispositifs inconnus connectés à l'appareil ou en «reniflant» le réseau en mode passif. Puis le mécanisme d'empreinte matérielle protège le standard de bus contre les attaques par usurpation de paquets. Ensuite, un détecteur d'anomalie basé sur le contexte utilise l'apprentissage machine non supervisée en temps réel pour détecter des messages anormaux grâce aux caractéristiques du voltage d'un bus.

Pour finir, l'engin d'explication d'anomalie se charge d'expliquer aux parties prenantes les anomalies précédentes détectées. Il commence par traduire les données machines en anglais

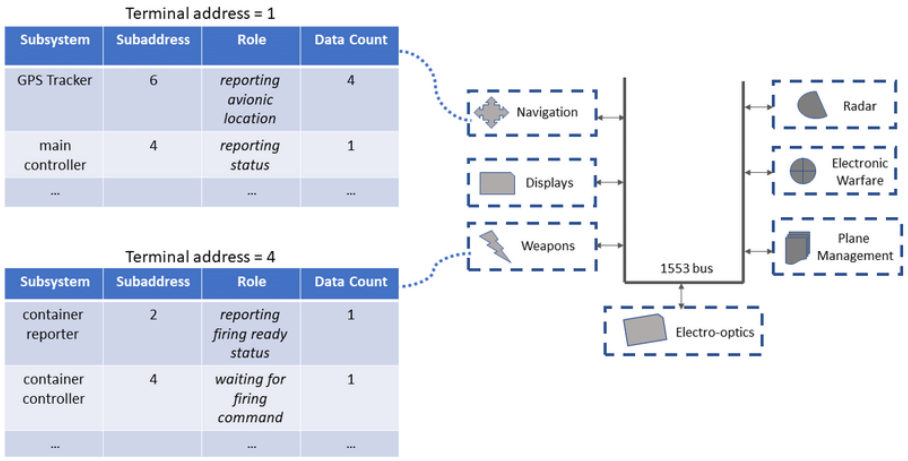


FIGURE 3.3 Exemple de *mapping* de sous-systèmes du 1553 sous AnoMili [15].

sous un modèle LSTM. L'engin d'explication va ensuite modéliser les alertes et les actions adéquates. On y lie des informations sur les vecteurs d'attaques et la description de l'attaque détectée. Sommairement, AnoMili présente un haut taux de détection de 99,45 % avec un faible taux de faux positifs. Il garde un niveau de calcul acceptable pour être une solution temps réel. Son mécanisme d'empreintes demande toutefois des changements physiques dans le matériel informatique, ce qui pourrait être complexe et coûteux à réaliser pour les manufacturiers. La modélisation (ou *mapping*), étant arbitraire, bénéficierait de NLP pour améliorer sa cohérence et sa standardisation.

3.2.4 Sommaire : Des contre-mesures en évolution

À ce jour, même si peu d'études existent encore dans le domaine de sécurisation des bus, le chiffrement, l'empreinte matérielle et les IDS sont ce qu'il y a de plus courant et d'efficace dans sa cyberdéfense. Nous dénotons des résultats plus prometteurs pour les solutions ayant recourt à l'intelligence artificielle et combinant plusieurs contre-mesures. L'apprentissage machine a été critiqué pour son approche probabiliste qui tend à offrir de faux positifs et des résultats peu uniformes. Toutefois, cette revue a présenté des travaux [9, 15] qui ont su démontrer des solutions d'IA avec des taux de réussite élevés. De surcroit, le pare-feu, la détection de logiciels malicieux n'ont pas été retenus puisqu'il demande d'adapter l'OS et le protocole de communication [34, 53]. Changer des composantes risque d'entrainer un énorme coût qui n'est pas viable.

En parallèle, on s'accorde aussi pour dire que l'ARINC 429 a des lacunes sous les trois

critères de sécurité. Les messages étant disponibles publiquement sur chaque composante, on ne répond pas à la confidentialité. Pour l'intégrité, certaines mesures qui limitent le nombre de transmetteurs peuvent être contournées [51]. Quant à la disponibilité, les collisions entre des mots sont possibles si un groupe malicieux décide d'envoyer un message au moment opportun. Enfin, le protocole ne satisfait pas non plus d'autres critères comme la non-répudiation et l'authentification. Un individu une fois infiltré dans le réseau du bus peut invalider d'autres composantes, s'ajoutant qu'il n'y a aucune mesure pour identifier l'avionique [54].

Nous notons tout de même que l'état de l'art présente une carence sur deux aspects qui méritent des recherches approfondies. D'une part, informer les parties prenantes, comme un pilote ou une équipe de maintenance, de la présence d'une cyberattaque n'est pas une tâche évidente et mérite qu'on s'attarde davantage à rédiger une «lingua franca». L'apprentissage machine est une technologie ayant offert de meilleurs résultats que nous le pensions en ce qui concerne les IDS. Pour standardiser un domaine, on argumente que cette approche probabiliste n'est pas adaptée face à l'inconsistance que pourrait retrouver un entraînement par rapport à un autre. Face au défi d'uniformisation des connaissances, des efforts de certification non négligeables peuvent remettre en question la viabilité de l'IA, du moins sur le long terme. D'autre part, les contre-mesures s'attardent davantage aux caractéristiques matérielles des paquets plutôt qu'à leur logique elle-même, de quoi contourner les mesures de sécurité et ouvrir la voie aux attaques sur la couche d'application ou aux attaques sur la chaîne d'approvisionnement. La section suivante décrit comment les ontologies offrent un apport à l'aviation et à ces dernières remarques.

Attaques	**Vecteur(s) d'attaque**	**Bus**	**Contre-mesure**	**Source(s)**
Usurpation de paquet Écoute clandestine	FMS Depuis l'ACAR ou l'ADS-B	N/A	N/A	[45]
Homme-du-milieu	EFB	ARINC 429 ARINC 717	Protection logiciel	[23, 48]
Déni de service sur l'AISD	EFB	N/A	Limiter le débit	[48]
Usurpation d'identité	IFE	N/A	Sécuriser le logiciel et les communicationS	[23]
Injection de code sur le PFD	ACARS (Signal VHF ou SATCOM)	ARINC429	N/A	[23, 47]
Attaque par homme du milieu Injection de code Chaîne d'approvisionnement	Mise à jours malicieuses sur un FLS depuis un CD-ROM ou l'Internet	MIL-STD-1533	IDS basé sur l'apprentissage machine usant des caractéristiques des signaux et des paquets	[3, 34]
Attaque technicienne avec un 2e transmetteur	LRU (ARINC 429)	ARINC 429	Empreinte matériel des caractéristiques électrique basé sur l'apprentissage machine	[6]
Usurpation de paquet de l'AHRS	CAN BUS	CAN BUS	N/A	[28]
Attaque par rejeu sur le PFD	CAN BUS	CAN BUS	N/A	[28]
Déni de service Usurpation de paquets	LRU (ARINC 429)	ARINC 429	Chiffrement	[51]
Déni de service Usurpation de paquets	Routeur Passerelle	ARINC 429 AFDX	IDS réseau basé sur les exigences	[8]
Usurpation d'identité Déni de service Injection de code	ADS-B	N/A	IDS ontologique basé sur les contraintes	[40]
Usurpation de paquet	Bus compromis	MIL-STD-1553	IDS analyseur de paquets et de signaux basé sur l'AI	[15, 43]
Déni de service Usurpation de paquets	Bus compromis	MIL-STD-1553	N/A	[30]
Usurpation de paquet	Depuis le GPS ou l'ADS-B	ARINC 429	N/A	[7]

TABLEAU 3.1 Sommaire des attaques, de leurs vecteurs et des contre-mesures proposées dans la littérature.

3.3 Revue des travaux sur l'utilisation de l'ontologie dans l'aviation

Cette section présente les travaux pertinents qui traitent de l'utilisation d'ontologie pour la sécurité.

3.3.1 Ontologies dans la cybersécurité

L'idée d'utiliser des ontologies en cybersécurité ne date pas d'hier. Dès 2003, Undercoffer et al. [55] démontrent leur utilité dans les communautés des IDS. On y modélise notamment des concepts appliqués à un réseau ou un site web pour détecter les attaques de DOS et de débordement de tampon. L'ontologie devient ainsi un système expert viable pour des fins de détection [10].

Par rapport aux bus, [56] introduit les relations sémantiques entre l'architecture cyberphysique Véhicule-à-Infrastructure (V2I). L'ontologie identifie les vulnérabilités qui peuvent impacter un système cyberphysique, et infère une contre-mesure de mitigation appropriée pour concevoir un système plus résilient. Leur solution est adaptée contre l'usurpation, le DOS et la contrefaçon de données. Un exemple de scénario mentionné est l'envoi de fausses données vers un feu de circulation pour changer sa couleur et créer un bouleversement sur la route. Même si l'ontologie n'est qu'en phase de prototype, elle a l'avantage d'être de bas niveau et se révèle idéale comme inspiration pour combiner le domaine des couches cyberphysiques d'un moyen de transport et de celui de la cybersécurité. La relation de ces deux domaines sera aussi étudiée par Adbelkader et al. [57] qui présentent OnSecta, un modèle d'ontologie sur la vérification et la validation de processus de sureté dans l'automobile. Le but est de contrer les potentielles menaces injectées depuis des bus automobiles avec fils ou sans fil tel le bus CAN ou l'Ethernet. Même si OnSecta n'est pas adapté à l'architecture avionique, la réussite de sa modélisation (*mapping*) entre la sécurité et l'automobile ouvre l'horizon quant à son application pour d'autres moyens de transport.

3.3.2 Modèles traditionnels de données dans l'aviation

Les modèles de données sur l'aviation traditionnels sont en quelques sortes les ancêtres des ontologies dans le domaine de l'ingénierie des connaissances. Ils ont été développés pour caractériser et gérer les données générées, utilisées et entreposées par une entreprise. Les dirigeants de l'industrie de standardisation de ces données interopérables sont : Organisation européenne pour la sécurité de la navigation aérienne (EUROCONTROL), la FAA et l'International Air Transport Association (IATA). Trois modèles importants ont été co-développés par ces organisations. Flight Information Exchange Model (FIXM) captures les données reliées aux vols

depuis les cycles de vie d'un vol : prévol, décollage, croisière et atterrissage. Aeronautical Information Exchange Model (AIXM) décrit l'espace de vol, ses routes, ses procédures, ses limites et son aide à la navigation. Weather Information Exchange Model (WXXM) modèle ses observations météorologiques, ses prévisions et ses mesures.

L'harmonisation de ces trois modèles est un but désirable dans la réalisation d'une application d'aviation interdomaine. Cela s'est toutefois heurté sur diverses barrières ces dernières décennies. Tout d'abord, les modèles de données souffrent d'un manque d'interopérabilité. Différents fournisseurs de données emploient des modèles différents, ce qui augmente les problèmes d'échanges. L'encodage, le nom, la sémantique, la résolution spatiale ou temporelle ainsi que les systèmes de mesures peuvent venir impacter cet échange. Par exemple, deux compagnies aériennes peuvent encoder les identifiants différemment. Notamment, des identifiants provenant de l'IATA vont contenir un code à trois lettres alors que ceux de l'OACI vont en contenir quatre. Cela a emmené les autorités aériennes à migrer du côté des ontologies. Pour continuer, une autre barrière aux modèles d'échange est qu'il souffre d'un «héritage biaisé». En d'autres mots, ces modèles ont été principalement créés pour faciliter les échanges entre des données de produits d'aviation préexistant. Ils n'ont ainsi pas été suffisamment généralisés pour servir aux produits de nouvelles générations. Une dernière préoccupation est le manque d'agilité et de maintenance de ces modèles. La plupart ont été définis en langage Unified Modeling Language (UML), ce qui les rend surtout encombrants et peu agiles. En effet, l'UML spécifie le modèle des dites données du langage et doit être implémenté avec un langage qui cible la représentation de celles-ci, tel le langage XML. Tenter une telle synchronisation apporte une épine de logistique sous le pied [36].

Face aux lacunes du modèle de donnée, Single European Sky ATM Research (SESAR) et Next Generation Air Transportation System (NextGen) ont avancé l'idée d'utiliser l'ontologie comme un pont pour intégrer plusieurs systèmes et sources d'informations ensemble afin d'améliorer l'interopérabilité, la performance, la fiabilité, la sécurité et la diminution des erreurs [20].

3.3.3 Ontologies courantes dans l'aviation

La proportion croissante et non négligeable d'ontologies en aviation a attiré l'attention des chercheurs. On présente ici les plus communément utilisées pour des prototypes. Les comparer peut sembler de prime abord une tâche infructueuse, vu la multitude de sous-domaines. Il est tout de même important de cerner leur applicabilité.

Le projet The Intelligent Semantic Query of Notices to Airman (ISQ NOTAM) conseille aux pilotes d'établir ou de changer les procédures de vol, les installations portuaires, les services

de vol ou d'autres opérations hasardeuses. Il est basée sur les normes de commissions des États-Unis et de l'Union européenne. En ce qui a trait à la FAA, on propose la *FAA Entreprise Information Management*, une ontologie standardisant les terminologies utilisées dans leur organisation dans le but de faciliter la recherche de documents et ultimement améliorer la prise de décisions [36]. Une autre ontologie, Unified Cybersecurity Ontology (UCO), est une extension d'un IDS et intègre différents schémas afin d'obtenir des données reliées à la cybersécurité. Disponibles sur GitHub, ses six concepts sont ceux de la vulnérabilité, de l'attaque, de la menace, de la conséquence, de l'atout et de l'attaquant [11].

L'ontologie ATMONTO, développée par la NASA, se démarque des autres ontologies par sa capacité à unir plusieurs concepts de l'aviation et a montré son extensibilité en intégrant davantage de concepts au fil des dernières années. Ces avantages compétitifs expliquent notamment pourquoi d'autres ontologies choisissent d'importer ATMONTO dans leur propre domaine de connaissance. Avec au moins 150 classes et plus de 100 propriétés, elle décrit les communications entre les pilotes et les tours de contrôle, incluant surtout des composantes de haut niveau, tels l'infrastructure de l'espace aérien (aéroports, manufacturiers), les éléments de vols, l'aide à la navigation des conditions d'opération de vol et des conditions météorologiques [11]. Contrairement à ISQ NOTAM qui se concentre autour de la localisation, des routes et des canaux de communication, ATMONTO [11] aborde plutôt l'aspect des composantes et les systèmes de l'avion. La NASA aspire à uniformiser son ontologie à grande échelle. On préconise ainsi d'importer ATMONTO dans notre ontologie.

L'ontologie ATMONTO est importée par Stefanidis [58] sur ICARUS, une ontologie à multiples paliers sur l'aviation qui facilite la description sémantique de domaines hétérogènes. ICARUS intègre aussi diverses couches de ressources d'informations à travers une plateforme du même nom. Son paradigme repose sur la combinaison de plusieurs ressources hétérogènes et sa tentative de pallier le manque de standardisation dans l'industrie. En combinant ATMONTO avec une ontologie épidémiologique liée à la virulence de la COVID-19 dans le monde, ICARUS infère, à travers des requêtes SPARQL, les avions qui seraient les plus susceptibles d'être infectés. La plateforme ICARUS n'est par contre pas encore disponible et seul le code des premières versions d'ICARUS est à source ouverte.

3.3.4 Ontologies liant la cybersécurité à l'aviation

Louis Philippe-Morel [53] présente un système de détection pour le contrôle aérien basé sur les ontologies pour entreposer les données ainsi qu'un moteur de requêtes ontologiques en langage SPARQL pour la détection d'anomalies. Ce détecteur va déceler notamment des manipulations malicieuses de signaux, dont des positions GPS insérées sur le trafic réseau.

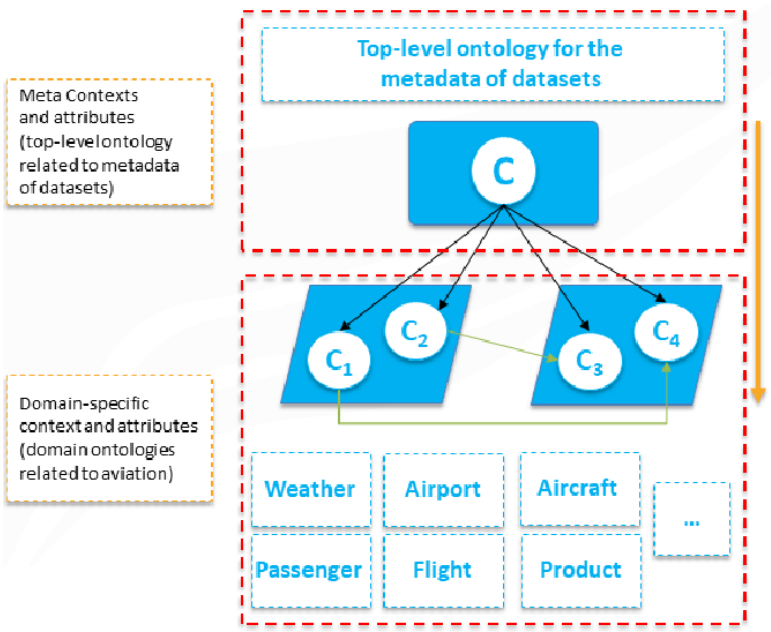


FIGURE 3.4 Approche multicouche de l'ontologie ICARUS où l'ontologie C de haut niveau intègre des domaines indépendants de l'aviation [58].

Il évite toutefois d'incorporer d'autres sémantiques existantes et choisit simplement de s'en inspirer. Cette décision risque d'engendrer des défis d'extensibilités dans l'avenir.

En 2018, l'ontologie Avionics Analytics Ontology (AAO) est proposée pour améliorer la prise de décision difficile pour un pilote dans le domaine de l'avionique analytique et des services ATM/UTM [59]. Elle aborde notamment la taxonomie d'un appel de détresse avec des données reliées à la météo, aux vols et à l'espace de l'air. Un de ses buts est aussi d'offrir un standard pour toutes ces différentes données. Elle incorpore le cadriciel UREF qui définit le concept d'incertitude. Malgré son interopérabilité avec UREF, AAO n'apparait pas être facilement intégrable à d'autres ontologies comme ATMONTO. Des travaux d'unification avec cette dernière sont à ce jour en cours. En 2021, l'équipe de recherche propose cette fois d'appliquer l'IA sur AAO pour supporter les tâches de prise de décision à travers deux scénarios [60]. Le premier évalue les situations dans l'espace aérien et le second l'autonomie dans l'évitement de collisions. Dans une recherche ultérieure, les chercheurs intègrent un raisonneur ATM dans leur recherche afin de réduire davantage la charge de travail des Contrôle de la circulation aérienne (ATC) et des pilotes [12].

C'est le projet SATIE qui réussit dans son architecture à lier une ontologie sur l'ingénierie de l'aviation et une autre sur la cybersécurité. Il s'agit d'ATMONTO et d'ICO. On y vise le développement d'une boîte à outils holistique, interopérable et modulaire pour faire face aux

menaces cyberphysiques dans les aéroports [11]. SATIE définit comment des événements et des alertes sont déclenchés dans l'avion, pour renforcer la création de solutions de sécurité. Sa fonction tient toutefois plus à la surveillance d'un aéroport plutôt que celui des bus.

Ayesha et al. [40] proposent d'utiliser une ontologie sur le ATC pour étendre le domaine de connaissance des IDS basés sur des contraintes. Cette approche tente de valider les contraintes de bas niveau définies sur les données applicatives, tel la validité des vitesses de vol. Des règles et des requêtes SPARQL sont développées avec les données pour représenter les menaces détectables et enrichir de la même façon l'ontologie. L'ADS-B est adopté comme vecteur d'attaque où trois modèles de menace sont implémentés : les avions fantômes, le viol de loi de la physique et l'usurpation de la localisation.

Dans son mémoire, De Miccieli [10] utilise une base de données ontologique, ATC-Sense, pour développer un système expert comprenant des règles de détections d'attaque sur l'ADS-B basées à propos de concepts de haut niveau du trafic aérien. Ses scénarios d'attaques quant à eux sont ensuite établis par l'identification des principales menaces. On retrouve comme attaques : le brouillage des communications, l'usurpation de l'identité et des paquets GPS et la contrefaçon de paquets. Les limites de cette approche sont qu'elle ne démontre pas de résultats concluants avec le brouillage et sa zone de couverture, étant expérimentale, peut créer de faux positifs dans une situation réelle. Même si notre expérience relève plus de l'explication d'anomalie plutôt que de sa détection, il demeure quand même pertinent dans notre expérience de nous assurer de poser des scénarios d'attaques avec divers facteurs sur les attaques que nous chinoiserons de couvrir.

3.3.5 La maintenance

Palacios et al. [61] proposent d'implémenter une ontologie dans le domaine de la maintenance et de produire un raisonnement automatisé. Contrairement au domaine de la cybersécurité, où l'on étudie des attaques, cette fois c'est les caractéristiques d'une défaillance et les réparations nécessaires qui sont dirigées vers les parties prenantes. On évite ainsi d'avoir recours à une tierce personne chargée d'une analyse qui pourra être longue, coûteuse et physiquement indisponible.

Les chercheurs font aussi remarquer que pour produire une ontologie sur l'aviation, il est mieux d'offrir un modèle conceptuel et de le nourrir ensuite de données. L'approche ascendante (*bottom-up*) des modèles d'échange de données est aussi décrite comme difficilement réalisable à cause de l'hétérogénéité et la récursivité des données qui produisent des récurrences.

3.3.6 Sommaire : Une ontologie des bus pour uniformiser l'avionique

Somme toute, intégrer le domaine de l'ontologie sur l'aviation est une tâche avec un intérêt démontré. L'essor des cyberattaques dans l'aviation ravive la pertinence d'utiliser une ontologie pour communiquer les menaces aux parties prenantes et s'adapte conjointement aux objectifs de maintenance [61] et de sûreté [11].

L'avionique fonctionne, comme on le rappelle, sous forme d'applications ou de services boîtes noires. Cette obfuscation limite l'aperçu de l'interaction entre le pilote et son environnement. Peu de systèmes avioniques savent ainsi s'adapter à ceux d'autres manufacturiers ou subir des améliorations [9]. Pour contrer ce défi, l'ontologie de notre EEA préconise comme structure de définir un sous-réseau de bus avionique standardisé, qui collectionne et analyse les données d'individus. Capable d'informer les parties prenantes des attaques informatiques par des inférences logiques, elle améliore ainsi la prise de conscience de l'humain sur les systèmes de vols critiques tout en liant des composantes avioniques de manufacturiers hétérogènes. Notre EEA, elle, s'adapte enfin aux stratégies d'affaires qu'un client souhaiterait tout de même préserver [7], puisque l'ontologie répond aussi au critère d'extensibilité.

CHAPITRE 4 REVUE DES MÉTHODOLOGIES

La sécurité de l'avionique et les ontologies de l'aviation ont reçu peu d'attention durant cette dernière décennie. Quelques acteurs ont su apporter des méthodologies qu'on juge intéressante dans le développement d'une ontologie apte à s'intégrer avec un EEA. On retrouve MITRE, Sesar, Next Gen ou Peroni [16,18,35]. Ce chapitre survole les méthodologies jugées pertinentes afin de concevoir notre EEA.

4.1 L'intérêt de la corporation MITRE

La corporation à but non lucratif axée sur la défense, MITRE, œuvre dans la conception d'ontologies et de cadres (ou *framework*) pour faciliter leur développement. Elle a été ces dernières années active dans le domaine de la défense. MITRE maintient deux dictionnaires de classification d'attaques qui sont le CVE (Common Vulnerabilities and Exposures) et le CWE (Common Weakness Enumeration). On y retrouve aussi la classification de moyens d'attaque CAPEC et un langage XML structurel STIX pour représenter les cybermenaces. MITRE argumente aussi pour la construction d'un langage commun et la définition d'un ensemble de concepts pour une compréhension partagée de la cybersécurité. Bref, MITRE insiste sur le besoin d'un vocabulaire contrôlé et d'ontologies pour assurer un progrès vers une science de la cybersécurité [16].

Leo et al. [16], chercheurs chez MITRE, présentent dans leur étude une méthodologie pour développer une ontologie afin de décrire les logiciels malicieux. Leur but premier y est la création d'une ontologie générique sur la cybersécurité exprimée en OWL. Leur paradigme repose sur la définition de quatre piliers : la victime, l'infrastructure, l'acteur malicieux et la capacité d'attaque. Les auteurs avancent une série de recommandations importantes. Tout d'abord, si le nombre de classes et de propriétés incorporées dans une ontologie devient trop grand, il est pertinent de considérer l'importation d'une ontologie déjà créée et d'établir des relations d'équivalences avec la nôtre. Enfin, pour mieux nous aider à définir une question de compétence prochaine, on doit se demander si notre approche sera ascendante ou descendante. Une analyse ascendante (*bottom-up*) requiert une compréhension de la sémantique des sources de données à intégrer. Quant à l'analyse descendante (*top-down*), on priorise plutôt la compréhension de la sémantique de l'utilisateur qui va utiliser au final notre ontologie.

Un autre papier du MITRE [14] présente la recherche et le développement de MITRE D3FEND. C'est un cadre sous la forme d'un graphe de connaissances de plusieurs contre-

mesures en sécurité le tout de manière précise et sans ambiguïtés. Cette ontologie se base sur l'idée que la cybersécurité doit pouvoir comprendre le problème qu'on essaye de résoudre, comment il a été résolu par le passé et enfin pourquoi une certaine solution serait meilleure. D3FEND établit alors un modèle d'ontologique de contre-mesures à faible granularité en OWL DL. Il utilise pour cela le cadre MITRE ATT&CK qui représente les Techniques Tactiques et Procédures (TTP) de l'attaque en utilisant les concepts définis dans D3FEND. Une contre-mesure est définie comme une classe et contient plusieurs entités. On retrouve dans la contre-mesure : sa définition, son fonctionnement, ses potentielles mises en garde d'implémentation, ses relations avec d'autres artefacts tels des opérations d'attaque sous ATT&CK ou bien d'autres contre-mesures. Cette ontologie n'est encore qu'à ses balbutiements. Ses résultats ont quand même su montrer que D3FEND est apte à appréhender le Web des données avec des brevets en cybersécurité. De plus, la solide réputation du MITRE fait d'elle une arme de défense solide à ne pas ignorer.

Le cadre du MITRE est aussi étudié dans la cybersécurité de l'aviation. Habler et al. [62] produisent une taxonomie en profondeur standardisant la connaissance et la compréhension de la cybersécurité de l'avionique. L'équipe intègre la méthodologie de MITRE ATT&CK depuis le point de vue de l'adversaire en étudiant son profil psychologique. Cette taxonomie sert ensuite à l'implémentation d'une ontologie qui définit les menaces et les acteurs qui font part des diverses attaques analysées. Le cœur de cette ontologie se situe néanmoins davantage sur l'analyse de menace plutôt que l'avionique en elle-même. Son utilisation de MITRE pour construire un profil psychologique demeure pertinente à intégrer selon nous dans ce mémoire. On argumente la nécessité d'appliquer ce cadre à la sécurité de l'avionique puisqu'il s'agit d'un domaine hautement complexe dû à ses connexions, ses composantes et ses surfaces d'attaque. Ce faisant, nous pourrons implémenter notre ontologie sous une modélisation standardisée.

4.2 Une méthodologie pour concevoir une ontologie

Des méthodologies préexistantes existent pour faciliter la conception d'une ontologie avionique. Le cadre ONTOCOR est un logiciel de développement basé sur les ontologies conçues par SESAR et NextGen [35]. Son expertise s'étend pour les systèmes de communication, de navigation et de surveillance. La philosophie d'ONTOCOR est d'unir des standards et des principes pour offrir une amélioration des techniques et des opérations de l'interopérabilité d'un aéronef et des systèmes ATM. Le cadre n'est toutefois qu'en phase théorique et n'a pas reçu de mises à jour depuis 2017. SAMOD (Simplified Agile Methodology for Ontology Development) [18] amène une méthode agile avec un processus de développement itératif pour développer une ontologie bien développée et documentée. SAMOD reste aujourd'hui couram-

ment cité, ce qui reflète son passage de l'épreuve du temps. L'auteur de SAMOD, Peroni, argumente qu'une ontologie qui veut tendre vers la perfection ne peut pas se construire en une seule fois, puisqu'il s'agit d'un processus long et laborieux. SAMOD se veut somme toute propice pour créer un prototype.

Hu et al. [63] introduisent RKMOF, un autre cadre qui vise la création d'une ontologie dédiée aux systèmes avioniques sans ambiguïté, sans incohérence et sans incomplétude. Son procédé évalue une modularité avec des métriques de cohésions et de couplages. RKMOF se révèle toutefois être un cadre trop sophistiqué pour la réalisation d'un prototype sur notre sujet de recherche et approprié dans le domaine des défaillances plutôt que des attaques.

4.3 Sommaire : Choix de la méthodologie

Somme toute, l'entreprise MITRE a su développer des outils adéquats pour préconiser la construction d'un cadre commun standardisé et avec une solide réputation pour une compréhension partage de la cybersécurité et de l'aviation. Le premier d'entre eux, est le cadre MITRE ATT&CK qui sera intégré afin de définir dans ce mémoire le *modus operandi* du profil des attaquants évalués et augmenter la pertinence de nos scénarios d'attaques. Le second outil sera d'intégrer à notre ontologie, MITRE D3FEND. En liant les concepts d'artefact à nos composantes avioniques, à nos scénarios d'attaque et aux contre-mesures, notre ontologie intègre un domaine interdisciplinaire sous un cadre standardisé.

Pour concevoir cette même ontologie avionique, la méthodologie de conception d'ontologie SAMOD sera utilisée en prenant une approche hybride du haut vers le bas (en anglais *top-down*) ainsi que du bas vers le haut (en anglais *bottom-up*). En approche du haut vers le bas, on s'assure que notre ontologie va intégrer adéquatement l'équipement avionique. Pour ce qui est de l'approche bas vers le haut, on s'assure de bien saisir la volonté des parties prenantes ciblées qui sont surtout le personnel de maintenance, le pilote et les ingénieurs en sécurité.

CHAPITRE 5 UNE NOUVELLE APPROCHE D'ENGIN D'EXPLICATION D'ANOMALIES BASÉ SUR L'ONTOLOGIE

Dans ce chapitre, nous expliquons la solution que nous avons implémentée. Nous décrivons le jeu de donnée, l'application de MITRE ATT&ACK ainsi que la modélisation de notre ontologie.

5.1 Jeux de données

Les données obtenues pour peupler la ABOX de notre base de connaissances proviennent de deux sources : la NASA et Bombardier.

5.1.1 Données de la NASA

Nous prenons les données de véritables vols extraits en juillet 2014. Les trajectoires de vols, les rapports météorologiques METAR, les registres des avions et les données d'infrastructures portuaires proviennent de la FAA, de l'OACI et du Bureau des Transports américains (DOT) [13]. La figure 5.1 présente un exemple d'instances sous ATMONTO intégrant ces données.

5.1.2 Données de Bombardier

Un de nos partenaires, le manufacturier Bombardier inc. nous a également fait part de données produites lors de simulations de vol avec certains paquets ARINC 429 formatés en paquets AFDX. La multinationale canadienne, centrée sur les avions d'affaires, capture les données de test du Bombardier Global Challenger 7500 sur un vol de 7 heures. Nous utilisons leur Interface Controle Document (ICD) pour produire des instances de paquets ARINC 429 pour la ABOX. On produit ainsi toutes les données possibles de désactivation d'un autopilote en statut normal. Les données de la simulation sont intégrées sans influence de notre part, en tant que chercheur. Elles sont ainsi produites à l'aveugle.

5.2 Application de MITRE ATT&ACK

Pour satisfaire notre objectif d'un engin standardisé, nous avons utilisé MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) (Voir Section 3) afin d'offrir

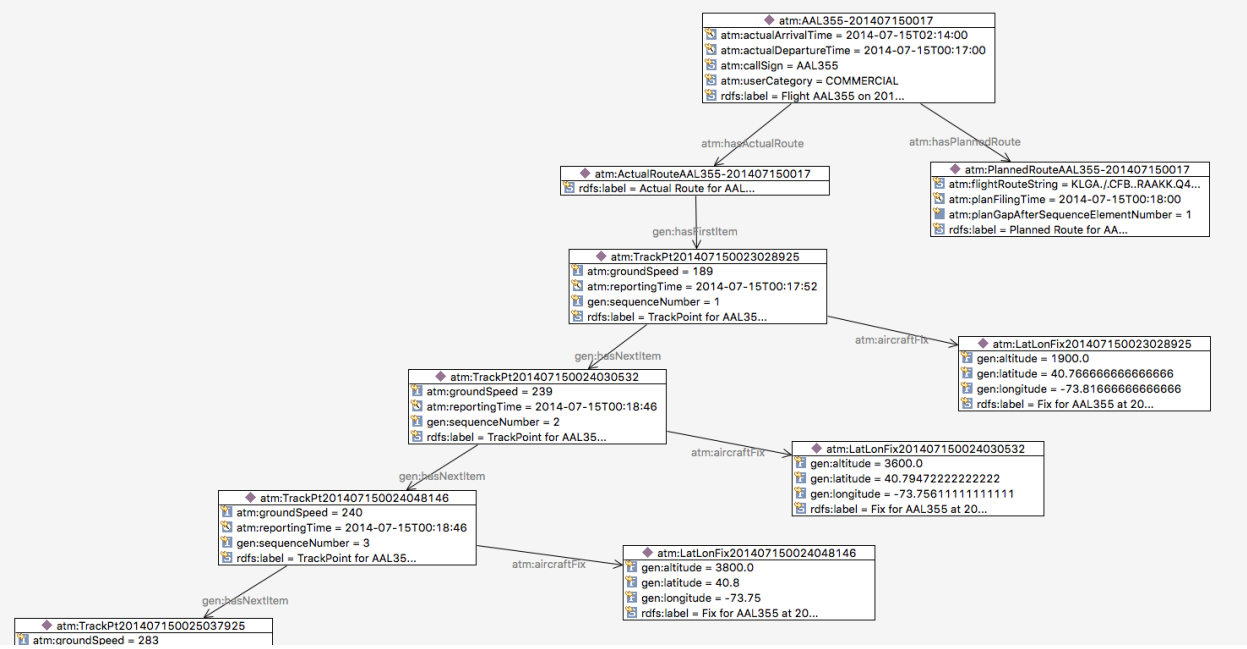


FIGURE 5.1 Structure d'un plan de vol planifié sous ATMONTO [13].

un cadre d'attaque standardisé. Par sa large adoption dans la communauté de la sécurité informatique et sa pertinence pour cartographier l'information, cette base de données va servir à modéliser nos TTPs (Techniques Tactiques et Procédures). Il génère une matrice d'attaque englobante pour ensuite expliquer les actions du point de vue de l'adversaire selon ses capacités, ses limitations et les impacts de ses attaques [62]. ATT&CK comporte les concepts de tactiques et de techniques :

— **Tactiques** : Représente le but de l'adversaire lors du lancement de sa technique d'attaque [62].
— **Techniques** : Représente la façon dont l'adversaire réalise son action pour atteindre son but. Les techniques peuvent être aussi divisées en sous-techniques qui représentent une plus large gamme d'actions spécifiques à l'attaque [62].

5.2.1 Les acteurs malicieux potentiels

Il existe divers acteurs pouvant chercher à s'attaquer à un avion. Pour modéliser sous MITRE ATT&CK un scénario d'attaque avec des acteurs pertinents pouvant causer des impacts considérables à l'aviation, nous observons les motivations, les moyens et les buts qui peuvent mener un individu ou un groupe à chercher à nuire à la sécurité d'un l'aéronef. Nous prenons en compte, dans notre analyse d'acteurs, celles précédemment réalisées [23, 62]. On subdivise les moyens d'un attaquant en cinq catégories : ses ressources disponibles, sa motivation, son

accès au système, ses connaissances en informatique et ses connaissances de l'avionique.

1. **Amateur** : L'individu possède des capacités limitées et n'est surtout motivé que par la notoriété. Il représente une faible menace.
2. **Criminel** : L'individu ou le groupe commet des activités malicieuses pour voler des données sensibles ou pour le profit. Pour un tel acteur, l'effort de s'attaquer à un avion est insuffisant à la vue de ressources matérielles limitées. Ceux-ci vont sûrement utiliser des COTS pour parvenir à leur fin.
3. **Hacktiviste** : L'individu ou le groupe est peu organisé et a en général comme but d'entraîner un accident, passer un message politique, assouvir une vengeance envers une personnalité publique ou réaliser un profit. On peut penser à des groupes écologistes extrémistes. Plutôt que d'attaquer un énorme aéronef hautement surveillé, cet acteur peut plutôt opter pour l'avion à réaction privé d'une cible spécifique. Certains garages sont peu surveillés [23]. Il demeure peu probable toutefois que cet acteur possède suffisamment de ressources malgré un accès physique pour changer des composantes ou réaliser une écoute clandestine. Les avions utilisant l'ARINC 429 sont souvent de petits jets souvent pour un public plus fortuné [23]. Même si le sabotage demeure limité, il peut quand même salir la réputation d'un compétiteur, augmenter les coûts d'un vol ou causer leur annulation. Vouloir nuire à cette classe de personne serait un objectif possible de la part de l'hacktiviste.
4. **Crime organisé et terroriste** : Le groupe ou l'organisation souhaite attaquer les services pour des raisons pécuniaires ou pour faire passer un message politique. Certaines actions pour saboter un avion peuvent être violentes, voire létales. On peut y retrouver des compétiteurs ou des organisations cybercriminelles qui sont connues pour extorquer de l'argent via le rançongiciel par le biais de COTS [23,62]. Les COTS doivent être obtenus depuis des manufacturiers approuvés et respectables durant la phase de test si l'un d'entre eux venait être corrompu. Toutefois, aucune spécification des COTS n'est définie par la FAA [25, 64]. Pour moins de 100 $ US, ce profil peut obtenir des logiciels ou du matériel informatique avionique pour réaliser des attaques. Enfin, par rapport à tous les accidents fatals de vols commerciaux ayant eu lieu entre 1950 et 2009, 9 % d'entre eux sont dus à un sabotage ou une altération de l'avion [23].
5. **État-nation** : L'organisation est financée par un état et vise à produire des attaques sur le plus grand nombre de cibles qui sont choisies avec minutie. Leurs ressources humaines et matérielles sont de grande qualité et parfois de nature cybermilitaire. Il est de plus à consacrer un temps considérable. Les buts incluent le dommage financier, l'obtention d'information d'affaire ou opérationnelle et la démonstration de peur. Il

peut recruter plus facilement du personnel mal intentionné (pirate *Black Hat*) comme un pilote, un membre de l'équipage, un passager, un membre du personnel de maintenance ou un manufacturier de COTS. Un tel personnel pourrait par exemple accéder physiquement aux dispositifs externes, aux capteurs et aux LRU afin de les endommager ou les manipuler. En cas de cyberguerre, les forces militaires seront prêtes de la cible et pourront même essayer de s'introduire dans les communications telles qu'il a déjà été le cas pour des incidents avec des drones prédateurs [23].

6. **Service d'intelligence** : L'organisation opère des troupes et acquiert des informations chez leur cible. Leur méthode inclut l'exploitation des vulnérabilités chez le système attaqué ou la surveillance de l'espace aérien. Tout comme l'État-nation, le service d'intelligence détient beaucoup de ressources matérielles, financières et humaines. Les attaques sur les drones sentinelles ou Stuxnet sont des exemples d'attaques menées par un service d'intelligence [23].

Le tableau 5.1 résume notre analyse d'acteurs pouvant potentiellement vouloir attaquer un avion. Parmi ces individus ou organismes, deux méritent d'être étudiés plus en profondeur en cybersécurité aérienne. On retrouve l'État-nation et les services d'intelligence. Puisque l'attaque d'un avion n'est pas une mince affaire, on évalue les amateurs, les criminels et les hacktivistes comme ayant des capacités limitées. Il en va de même pour le crime organisé ou les terroristes qui, en plus de cela, ne risquent pas d'obtenir un assez grand gain pour l'effort encouru. Toutefois, des adversaires financés par un état disposent d'un nombre important de ressources humaines et de matériels. On juge leurs capacités suffisantes pour représenter une menace dans l'aviation. Les cas d'ingérences dans des affaires d'État deviennent de plus en plus courants ces dernières années, notamment, dans la politique d'élection [65] ou dans le technonationalisme [66]. On s'attend à voir un intérêt dans l'aviation encore plus grand ces prochaines années. En cas de cyberguerre, les forces militaires pourront essayer de s'introduire dans les communications telles que nous l'avons mentionnées lors d'incidents en Irak [23]. Somme toute, notre analyse sous MITRE contre les attaques de cybersécurité aérienne va prendre en compte ces deux acteurs.

Acteur	Ressources disponibles	Connaissances en informatique	Connaissances en avionique	Accès au système	Motivation	Impact [R + Ci + Ca + A + M) / 5]
Amateur	1	2	1	1	1	1.2
Criminel	1	2	1	1	1	1.2
Hacktiviste	1	2	1	1	3	1.6
C. O ou Terroriste	2	3	3	2	3	2.6
État-Nation	4	4	4	3	4	3.8
Service d'intelligence	4	4	4	3	4	3.8

Tableau 5.1 Analyse de risque des acteurs potentiels.

5.2.2 Modélisation de la Matrice

Dans le cadre de notre recherche, nous adaptons le cadre du MITRE sur des scénarios d'attaques ciblant l'avionique. On reflète ainsi les phases d'attaques du cycle de vie d'un adversaire. Nous identifions les actions spécifiques de l'attaquant à travers les *techniques* et les catégorisons par *tactiques*.

Notre utilisation de la matrice MITRE ATT&CK Enterprise v12, lors de notre modélisation, est justifiée pour deux raisons. D'une part, certaines attaques relèvent de l'infiltration au sein d'une entreprise, ici d'un avion, ou de la chaîne d'approvisionnement. La version Enterprise est conçue pour les systèmes informatiques traditionnels et vise à fournir une cartographie complète du profil de l'attaquant. De ce fait, elle couvre un plus vaste éventail d'attaques. La version ICS a aussi été envisagée. Elle est spécifiquement conçue pour les systèmes de contrôle industriel et de processus tel des capteurs, des actionneurs et des réseaux industriels. Toutefois, son éventail est plus restreint. D'autre part, la version Entreprise est la seule version compatible avec MITRE D3FEND que nous utilisons pour modéliser notre ontologie. Pour obtenir des résultats efficaces et opérationnels dans notre preuve de concept, nous optons donc pour la version avec la meilleure compatibilité à ce jour.

Nous avons utilisé la matrice du MITRE pour identifier les TTPs les plus courantes pour l'attaque sur l'ARINC 429. En utilisant ces informations, nous avons ensuite développé deux scénarios d'attaques.

La figure 5.2 présente notre matrice avec chacun des TTPs potentiels notés sur une échelle de 0 à 100. Cette base de données et notre analyse d'acteur servent de fondation dans le développement d'un modèle de menace ou d'un outil de cyberdéfense. Nous produisons grâce au cadre, deux scénarios d'attaques qui seront utilisés pour réaliser trois cas d'études avec notre ontologie.

La Matrice contient 12 tactiques [67]. Toutefois, seules 9 d'entre elles forment le profil de l'attaquant. Les tactiques de *Persistence*, de *Privilege escalation*, de *Credential Access* et de *Lateral Movement* ont été jugées non adaptées au profil de l'attaquant. L'acquisition ou le transfert de privilèges ne sont pas nécessaires pour une technologie n'ayant déjà pas de mesures d'authentification et avec un réseau simple.

— ***Reconnaissance et Resource Development*** : L'adversaire collecte et acquière passivement des informations qui seront utilisées dans son attaque pour ces deux tactiques. Ce dernier pose des actions comme acheter un ICD de l'ARINC 429, collecter une banque de mots ARINC 429 en ligne ou surveiller les déplacements d'une cible. Pour avoir accès à l'avion, il cherche à s'y rendre physiquement ou bien obtenir une

autorisation officielle, voire officieuse.

— ***Initial Access*** : L'adversaire établit un premier pas vers la cible. Il rajoute directement du matériel informatique sur l'aéronef, établit des relations ou compromet la chaîne d'approvisionnement.
— ***Execution*** : L'adversaire exécute des actions après avoir établi son accès initial, tel exécuter un fichier malicieux.
— ***Defense Evasion*** : L'adversaire cherche à contourner la détection et les contrôles de sécurité. Pour ce faire, il masque sa présence en camouflant ses services, ses noms et ses tâches comme étant de confiance.
— ***Discovery*** : L'adversaire acquiert davantage d'information pour supporter ses objectifs. Cela consiste à rechercher davantage de vulnérabilité dans le système ou dans ses appareils périphériques.
— ***Collection*** : L'adversaire collecte des données qui l'aideront à réussir ses objectifs. Elles peuvent être obtenues depuis une avionique corrompue ou un appareil périphérique depuis un port.
— ***Command and control*** : l'adversaire contrôle ses opérations à distance depuis, par exemple, une avionique infectée.
— ***Exfiltration*** : L'adversaire extrait des données qu'il a acquises en évitant idéalement toute détection d'un comportement malicieux.
— ***Impact*** : L'adversaire affecte la confidentialité, l'intégrité ou la disponibilité des actifs après avoir atteint son objectif. L'impact peut consister à faire du *spoofing* ou une attaque DOS.

FIGURE 5.2 Matrice MITRE ATT& CK comprenant les attaques potentielles sur l'ARINC 429.

Scénario 1 : Ajout d'une composante avionique pour produire un cas d'usurpation (*spoofing*)

Le premier scénario, produit avec notre matrice, correspond à l'ajout dans l'avion d'un LRU malicieux. Il est ajouté par l'attaquant Spooner, un employé *Black Hat* faisant partie de l'équipe de maintenance. Le LRU voyou possède une vulnérabilité *zero day*, c'est à dire ne faisant pas partie d'aucune publication ou rapport. Son but est d'engendrer une attaque par usurpation de paquet en perturbant l'appareil, ce qui peut causer des dommages irréversibles aux composantes électroniques et physiques. Spooner a une connaissance technique du matériel et de la topologie du système. Cette attaque a été démontrée par Markevich et al. [6] et étudiée par la suite [7, 51]. On assume que Spooner utilise des COTS [8]. Enfin, les cas d'attaques d'usurpation sur les bus deviennent de plus en plus populaires [15, 28].

On établit certaines hypothèses dans notre étude :

- Spooner est financé par un service d'intelligence.
- L'avion ciblé correspond au jet d'un homme d'affaires laissé dans une aérogare et qui communique avec l'ARINC 429.
- Spooner opère au sol.

Nous décrivons désormais notre scénario en parcourant les techniques de notre matrice avec le plus fort impact. On se fie à notre légende. La figure 5.3 présente le scénario spécifique avec le cadre.

— ***Reconnaissance*** *Active Scanning — Wordlist scanning* : Spooner obtient de l'information sur les étiquettes (*labels*) ARINC 429 utilisées via la surveillance du réseau d'appareils avioniques.

— ***Ressource Development*** *Stage capabilities— Upload Malware* : Spooner introduit un composant malveillant sur le système avionique auquel il engendre son attaque, il prend alors le contrôle du composant et peut modifier les opérations de la transmission et de la réception des paquets ARINC 429.

— ***Initial Access*** *Hardware Addition* : Spooner ajoute son LRU corrompu à l'avion cible. Cette avionique devient le vecteur d'attaque et obtient accès au système.

— ***Execution*** *User Execution — Malicious files* : Spooner incite le LRU à exécuter des actions sur l'avionique. Ayant des connaissances en pilotage et comment l'avion opère, Spooner a une idée des actions qu'un individu entreprend.

— ***Defense Evasion*** *Masquerading — Match legitimate name or location* : Spooner masque la présence de son LRU en faisant correspondre des labels de mots ARINC 429 avec des entrées légitimes en s'assurant d'un bon *checksum* de parité, SDI et SSM.

— ***Discovery*** *System information discovery* : Le LRU de Spooner tente d'obtenir da-

vantage d'information sur l'architecture du réseau. On peut ainsi mieux déterminer le comportement d'attaque adéquat.

— ***Collection*** *Adversary-in-the-middle— DHCP Spoofing* : Le LRU de Spooner, ayant schématisé le réseau, intercepte, depuis son LRU corrompu, le flux de données circulant dans le trafic réseau. En prenant en compte ces données, il usurpe la réponse d'une des avioniques ciblées. Le LRU peut alors envoyer de faux messages en se faisant passer pour un transmetteur légitime.

— ***Command and control*** *Traffic signaling— Non-application layer protocol* : Pour contrôler certaines communications, le LRU de Spooner envoie des commandes depuis le LRU voyou et ses récepteurs, ce qui produit une réaction en chaîne dans le réseau.

— ***Exfiltration*** — *Over C2 Channel* : Pour une attaque passive, si Spooner peut récupérer son LRU, il peut en extraire des données avec les ports de réception.

— ***Impact*** *Data manipulation — Transmitted Data Manipulation :* En altérant la transmission des messages, Spooner tente de causer une attaque de *spoofing*, réduisant ainsi l'intégrité du système. On manipule le résultat des activités. Ceci peut causer des incidents, voire des accidents durant un vol. Une telle attaque peut quand même affecter la réputation et les affaires des parties prenantes de l'avion ciblé en plus de la prise de décision de l'équipage.

Scénario 2 : Chaîne d'approvisionnement de l'avionique compromis pour engendrer un déni de service (DoS)

Le second scénario produit avec notre matrice correspond à la manipulation du matériel informatique (*hardware*) d'un LRU lors du processus manufacturier. L'attaque est orchestrée par Denise qui souhaite engendrer une attaque DOS durant l'opération d'un aéronef. Ceci peut causer des dommages irréversibles aux composantes électroniques et physiques. Elle a une connaissance technique de l'avionique sans toutefois connaître de topologie précise du modèle d'avion ciblé. Cette attaque a été démontrée par De Santo et al. sous le bus MIL-STD-1553 [30] et évaluée comme probable sur l'ARINC 429 [51]. On assume que Denise corrompt un appareil non-ATS [8]. Corrompre la chaîne d'approvisionnement peut possiblement déjouer les mesures de cybersécurité qui ne prennent pas en considération la couche application [9]. Il s'agit aussi récemment d'une technique jugée à risque dans le milieu de l'aviation [3, 8]. Enfin, les DOS sur les bus se font récemment de plus en plus nombreux [48].

On établit certaines hypothèses dans notre étude :

- Denise est financée par un État.
- Les avions cibles correspondent à des modèles ennemis qui communiquent avec l'ARINC 429

ou concurrents à cedit État.
- Denise opère sans avoir d'accès physique à l'avion.

Nous décrivons désormais notre scénario en parcourant les techniques de notre matrice avec le plus fort impact. On se fit à notre légende. La figure 5.4 présente des scénarios spécifiques avec le cadre.

— ***Reconnaissance*** *Searched Closed Sources — Purchase Technical Data* : Denise achète à travers des sources privées, potentiellement dans le marché noir, des informations sur ses cibles. Il peut s'agir de base de données, de balayage de mots ARINC 429 ou d'architectures électroniques, informatiques ou logicielles.
— ***Ressource Development*** *Stage capabilities— Upload Malware* : Denise introduit un logiciel malveillant sur un système avionique cible, il prend alors le contrôle du composant et peut modifier les opérations de la transmission et de la réception des paquets ARINC 429.
— ***Initial Access*** *Supply Chain Compromise— Compromise Hardware Supply Chain* : Denise manipule le côté *hardware* d'un LRU avant sa réception finale chez le client. Denise insère aussi des portes dérobées pour rendre difficile la détection. Cela lui offre un certain degré de contrôle.
— ***Execution*** *User Execution — Malicious files* : Denise incite l'usager à exécuter des actions sur l'avionique. Ayant des connaissances en pilotage, Denise a une idée des actions qu'un individu entreprend.
— ***Defense Evasion*** *Masquerading — Masquerade Task or Service* : Denise masque la présence de son LRU en faisant correspondre des groupes de mots ARINC 429 avec des entrées légitimes en s'assurant d'un bon *checksum* de parité, SDI et SSM.
— ***Discovery*** *Peripheral Device Discovey* : Après avoir été installé, le LRU de Denise obtient des connexions uniplex avec l'avion cible. Il peut alors schématiser les avioniques périphériques connectées entre depuis les récepteurs et les transmetteurs du LRU afin de trouver davantage de vulnérabilités «hôtes».
— ***Collection*** *Adversary-in-the-middle— DHCP Spoofing* : Le LRU de Denise, ayant schématisé le réseau, intercepte, depuis son LRU corrompu, le flux de données circulant dans le trafic réseau. En prenant compte de ces données, il usurpe la réponse d'une des avioniques ciblées. Le LRU peut alors envoyer de faux messages en se faisant passer pour un transmetteur légitime.
— ***Command and control*** *: Traffic signaling — Port Knocking* : Pour activer des ports, Denise configure son LRU pour cacher des ports de contrôle. Le LRU peut alors envoyer une série de tentatives de messages sous une séquence définie sur un port cible.
— ***Exfiltration*** N'ayant pas d'accès physique à l'avion, Denise n'effectue aucune action

d'exfiltration.

— ***Impact*** *Data manipulation — Direct Network Flood* : En inondant les transmissions d'un haut volume de messages, Denise tente de causer une attaque DOS, réduisant ainsi la disponibilité du système. Elle manipule ainsi le résultat des activités. Ceci peut causer des incidents, voire des accidents durant un vol. Même si l'attaque est répertoriée durant une phase de maintenance, une telle attaque peut quand même affecter la réputation et les affaires des parties prenantes de l'avion ciblé.

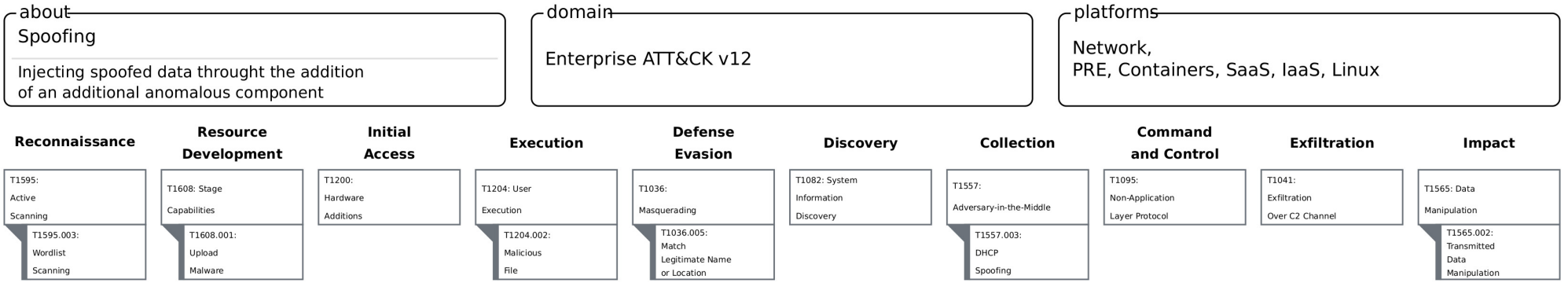


FIGURE 5.3 Matrice de l'usurpation de paquets à travers l'addition d'un LRU corrompu.

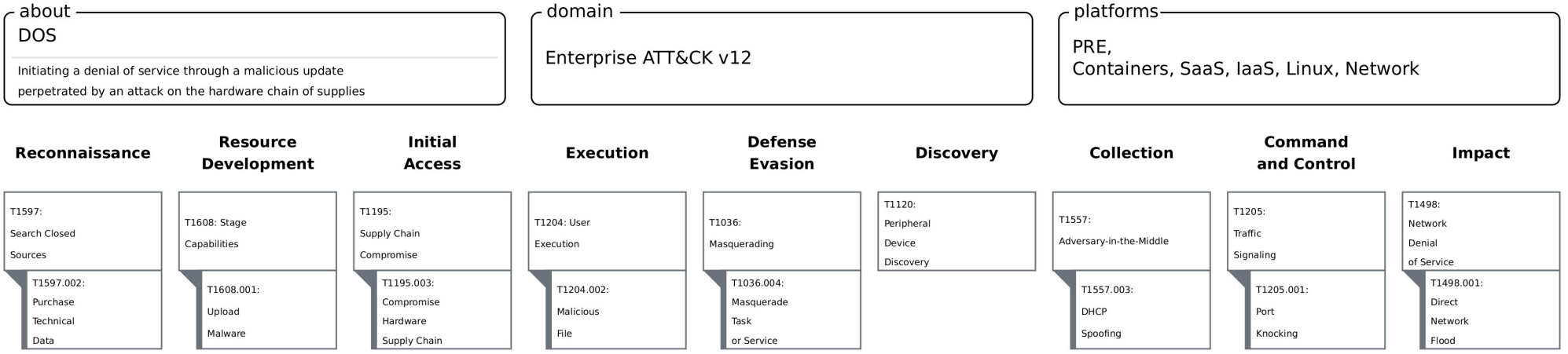


FIGURE 5.4 Matrice d'un déni de service à travers une attaque sur la chaîne d'approvisionnement.

5.3 Modélisation de l'ontologie

5.3.1 Onto-by-wire

L'implémentation de notre Engin d'Explication d'Anomalies intègre trois ontologies : *onto-by-429*, ATMONTO et MITRE D3FEND. Elles assurent une communication interdisciplinaire. L'ontologie *onto-by-429* représente le standard de bus ARINC429. Avec ATMONTO, on peut lier des messages ARINC 429 avec des vols complexes. Enfin, D3FEND relie les LRUs aux sous-classes d'artefacts et intègre les scénarios d'attaques et les sous-classes du concept de contre-mesure. L'ontologie complète porte le nom d'*onto-by-wire*. Elle compte au total 1667 classes, 340 propriétés d'objets et 247 propriétés de données. Un analyse est présentée en détail à la section 5.3.4.

Onto-by-429

Onto-by-429 (fbw) est une ontologie que nous avons créée. Elle compte au total 35 classes, 28 propriétés d'objets et 247 propriétés de données. La TBOX prend en compte des règles d'équivalence qui s'adaptent aux standards de l'ARINC 429 [27] pour préserver notre objectif de standardisation. On définit les concepts de champs de données ARINC, ses règles de validité, le LRU et la phase de vol. L'interopérabilité avec ATMONTO se fait en liant les concepts équivalents comme l'avion (*eqp : Aircraft equivalent-to fbw : aircraft*), en précisant les sous-classes (*fbw : RadioAltmeter is-a eqp : AircraftNavigationSystem*) ou en définissant des propriétés liant deux ontologies (*atm : AircraftTrackPoint fbw : hasCommunicationFix some fbw : FlyByWireFix*).

ATMONTO

ATMONTO a pour but d'intégrer des données hétérogènes en aéronautiques, le tout dans la volonté d'être utilisé dans l'investigation. ATMONTO est une pierre angulaire qui permet à notre ontologie de prendre en compte les multiples aspects d'un vol. Il combine les sous-ontologies de cinq sous-domaines chez *onto-by-wire* : l'équipement (Eqp), la gestion du trafic aérien (ATM), les données de vol (Data), les concepts des parties prenantes gouvernementales (Nas) et des concepts génériques à l'aviation (Gen) [12]. Intégrer ATMONTO à notre ontologie nous permet ainsi de préserver une consistance dans nos terminologies et d'agrandir la richesse de nos inférences.

MITRE D3FEND

L'ontologie MITRE D3FEND (d3fend) classifie et décrit les techniques de cyberdéfense de la corporation MITRE. Elle est compatible avec les TTP des scénarios que nous avons produits grâce aux classes et aux propriétés de MITRE ATT&CK (d3fend : ATTACKThing), intégrée dans D3FEND. ATTACKThing offre de plus les méthodes qui servent à compromettre les cibles. On offre ainsi à *onto-by-wire* des attaques obéissant à un autre standard. Nous l'intégrons afin de décrire le profil d'un attaquant, puis d'inférer les outils de cyberdéfense adéquats pour le contrecarrer (*d3fend : DEFENDThing*). L'interopérabilité d'*onto-by-wire* est réalisée en liant les actifs que l'attaquant cherche à compromettre (d3fend : Artefact) avec le matériel de l'avion (*eqp : RadioAltmeter* <u>*is-a*</u> *d3fend : DigitalArtifact*). On rajoute aussi une classe de scénario d'attaque (*d3fend : ScenarioATTACK*) et cinq nouvelles propriétés (*d3fend : next-ATTACK-scenario*, *d3fend-object-scenario-property*) adaptées à être intégrées dans nos scénarios.

5.3.2 Méthodologie SAMOD

Le processus de développement agile Simplified Agile Methodology for Ontology Development (SAMOD) [18] est appliqué (voir Figure 5.5) pour produire notre ontologie. L'approche itérative et incrémentale de SAMOD est adaptée pour notre projet de développement puisqu'il s'agit d'une ontologie de moyenne à petite taille. L'approche est flexible et rapide.

Nous divisons SAMOD en trois itérations de trois semaines chacune qui doivent produire un nouveau *modelet* accompagné d'un ensemble de *cas de tests*. Chaque itération est accompagnée d'un *scénario motivant* qui décrit le domaine que nous tentons de modéliser en une ontologie ainsi que des descriptions informelles et intuitives de la problématique. Il s'agit d'une ébauche de la sémantique intuitive prévue. La première itération produite par le *modelet* de l'ontologie *onto-by-429*.

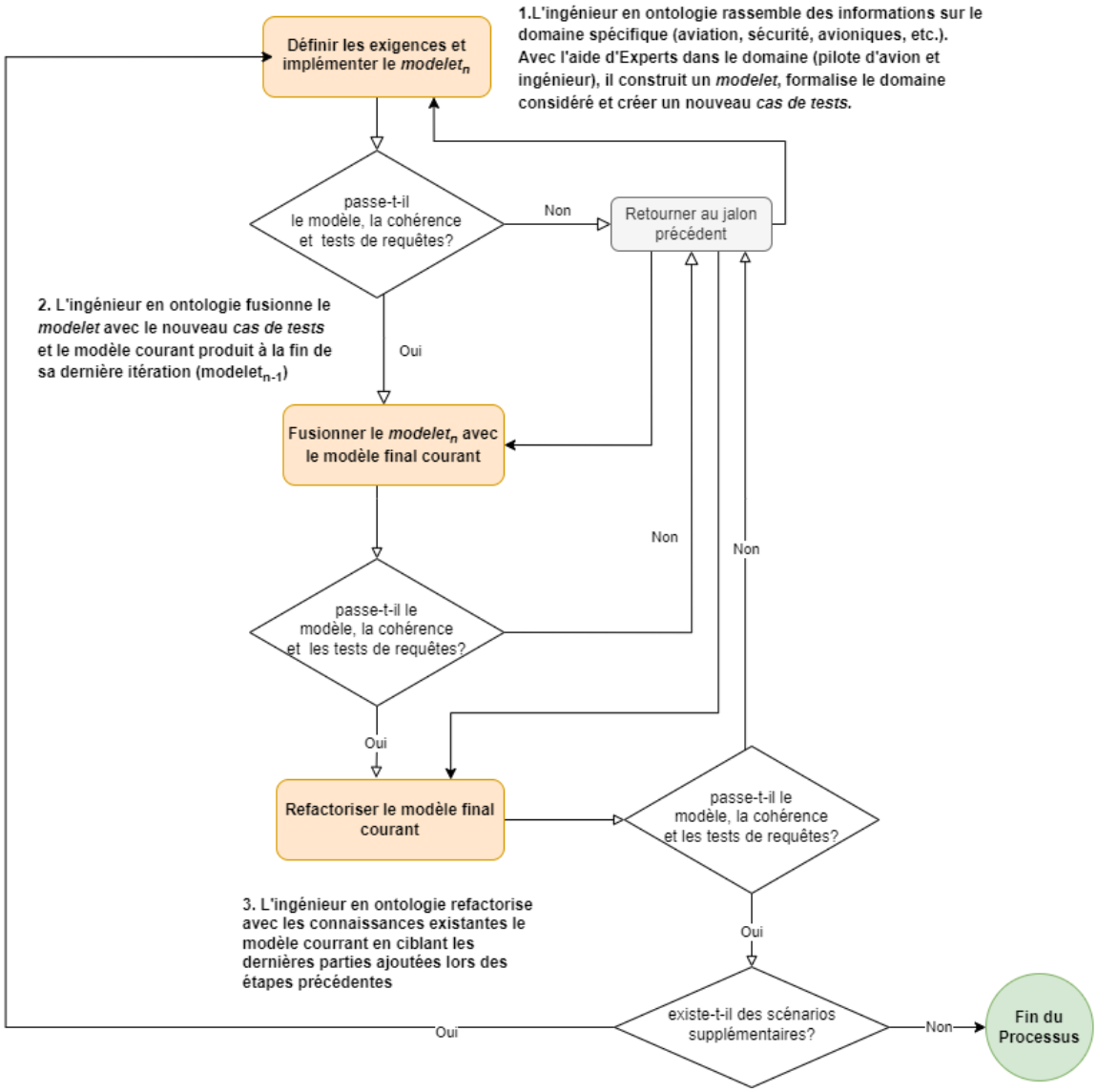


FIGURE 5.5 Application de SAMOD sur *onto-by-wire*.

On présente au tableau 5.2 le scénario motivant pour produire l'ontologie du système de bus ARINC 429. La seconde itération produite intègre *onto-by-429* avec ATMONTO. La troisième itération rajoute D3FEND dans le *modelet*. Chacune des itérations est validée par des autorités. L'EEA fait appel à un ingénieur spécialisé dans l'ingénierie des connaissances pour valider la structure de l'ontologie en observant notamment que l'ontologie évite des inconsistances. On fait aussi appel à un véritable pilote qui s'assure du réalisme de nos scénarios d'attaques et de la pertinence de nos questions de compétence. L'expert en ontologie, ici l'auteur de ce mémoire, demeure la seule autorité évaluant le modèle ontologique. Les tests sont présentés au chapitre suivant. Cette approche a pour but d'éviter le biais des partenaires ou du pilote en procédant à des expériences en double aveugle.

La formule à l'itération n d'un Cas de tests est [18]

$$T_n = (MS_s, CQ_n, GOT_n, modelet_n, ABOX_N, SQ_n) \tag{5.1}$$

où :

MS est un scénario motivant (voir tableau 5.2)

CQ est une série de questions de compétence (voir tableau 5.6)

GOT est un glossaire de termes pour le domaine

modelet est un modèle singleton qui décrit un domaine particulier

ABOX est un jeu de données ABOX, écrit en OWL, adapté à la TBOX courante

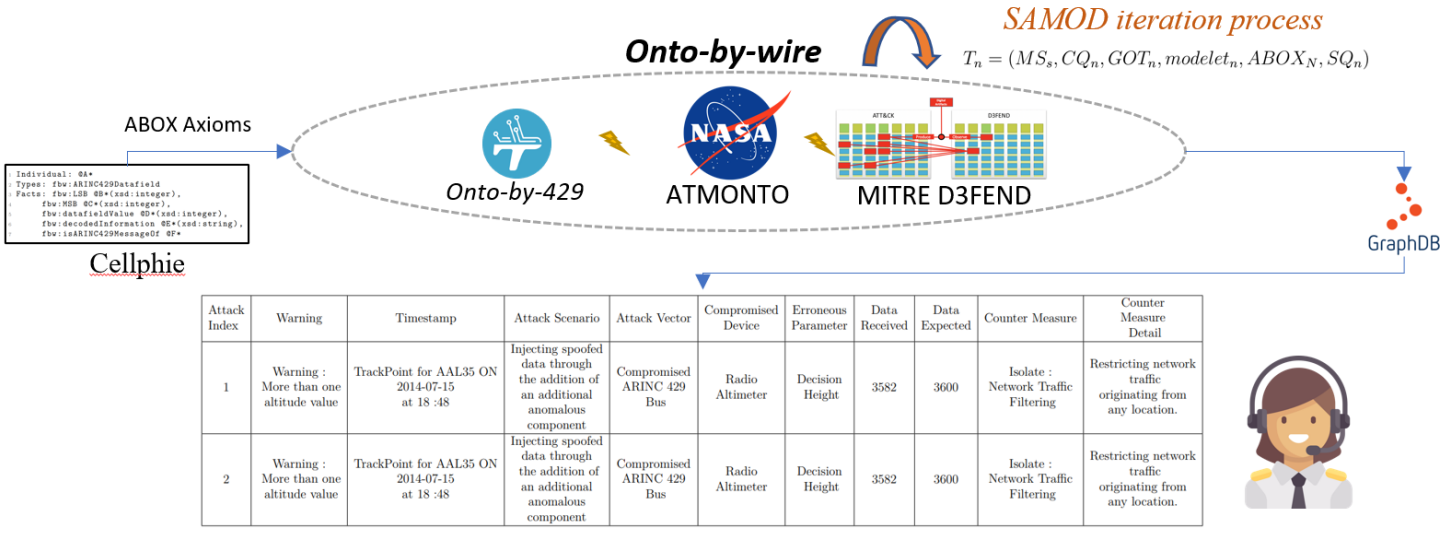
SQ est une série de requêtes écrites en SPARQL 1.1 (voir listing B)

Nom	ARIN429 de la TBOX d'une ontologie
Description	Les Spécifications ARINC 429 pour les avions commerciaux définissent un standard d'exigence pour le transfert de données digitales d'un système avionique à un autre. Dans une ontologie OWL, l'ARINC 429 apparaît dans la définition de classes, propriétés, types de données et d'individus. Pour ces entités, une description explicite de la nature et des caractéristiques de l'ARINC est requise. Analyser et décrire la nature de l'ARINC 429 en entité ontologique est une activité a priori objective puisqu'elle obéit à un standard. Toutefois, puisqu'une marge de manœuvre est autorisée par un transporteur, l'ontologie peut faire intervenir un peu de subjectivité. C'est pourquoi elle s'aide d'un Document de contrôle d'interface (ICD) pour assister à la création de sa TBOX. Le concept de « message ARINC429 » occupe une grande place dans l'ontologie de l'ARINC429. Elle doit comporter des champs de données propres à l'ARINC 429, occupant aussi la TBOX (label, parity, sdi, ssm, data). Ses sous-classes correspondent aux différents types de messages ARINC429 disjoints : BNR, BCD, Discrete. Des règles d'équivalences pour chacune des sous-classes sont précisées. Le champ de données SSM (Sign/Status Matrix) comporte une sous-classe pour chaque stratus possible d'un message. La TBOX décrit le côté unidirectionnel de l'ARINC429. Le bus d'une composante avionique est dotée d'une source et d'une destination. Ces bus sont liés à des LRUs (Line-Repleacable Unit) qui sont des sous-classes de ces composantes avioniques. Ils peuvent dans la plupart du temps transmettre et recevoir des messages. Le champ de donnée SDI indique aussi le nom de la source et de la destination du message. L'ARINC 429 peut être décrit avec un référent quantitatif ou qualitatif. L'aspect quantitatif concerne les limites des valeurs que peut prendre un mot ARINC 429. Ce dernier est formé de 32 bits et possède un label sur 8 bits avec une valeur décimale comprise entre 0 et 355. Un autre référent qualitatif est celui du bit significatif qui permet à l'ontologie d'inférer le champ de donnée ARINC429. Enfin, un message ARINC 429 ne peut comporter qu'un transmetteur pour au plus 20 récepteurs. Pour ce qui est du qualitatif, il concerne l'identification d'un discriminant auquel les limites d'une valeurs dans l'ontologie sont identifiables par un ICD. On retrouve par exemple la signification d'un SSM qui détermine la validité de certaines données (normal, erreur, sans données, test) ou bien des équivalences de classe. Une autre valeur qualitative est le «label» qui sert d'identifiant pour un message ARINC 429.
Test de consistance de la ABOX 1.1	L'ontologie génère des avions capables d'inférer s'ils sont commerciaux ou militaires selon le type de bus (AFDX, MIL-STD-1553), ARINC429)
Test de consistance de la ABOX 1.2	L'ontologie génère un avion commercial usant un bus militaire MIL STD 1553 et affiche une erreur d'inconsistance

TABLEAU 5.2 Exemple de scénario motivant sous la première itération d'*onto-by-wire* sous SAMOD

5.3.3 Environnement de développement

La conception de l'ontologie effectuée dans cette recherche a demandé la création d'un environnement de travail logiciel adéquat. La figure 5.6 représente un schéma d'un premier cas d'étude (voir sous-section 5.3.5) envoyé à notre l'EEA. L'ontologie de notre engin est peuplée d'instance. Des questions sont ensuite posées à l'ontologie qui offre, par la suite, ses résultats. Elle comprend les résultats du Tableau 6.3, disponible pour plus de clarté.



Attack Index	Warning	Timestamp	Attack Scenario	Attack Vector	Compromised Device	Erroneous Parameter	Data Received	Data Expected	Counter Measure	Counter Measure Detail
1	Warning : More than one altitude value	TrackPoint for AAL35 ON 2014-07-15 at 18 :48	Injecting spoofed data through the addition of an additional anomalous component	Compromised ARINC 429 Bus	Radio Altimeter	Decision Height	3582	3600	Isolate : Network Traffic Filtering	Restricting network traffic originating from any location.
2	Warning : More than one altitude value	TrackPoint for AAL35 ON 2014-07-15 at 18 :48	Injecting spoofed data through the addition of an additional anomalous component	Compromised ARINC 429 Bus	Radio Altimeter	Decision Height	3582	3600	Isolate : Network Traffic Filtering	Restricting network traffic originating from any location.

FIGURE 5.6 Représentation de l'Engin d'Explication d'Anomalies (EEA) [13, 14].

Le premier élément de notre environnement est Protégé. Développé par l'Université de Stanford, il s'agit d'un outil à source ouverte largement utilisé pour la modélisation et la gestion des ontologies. Protégé offre en effet des fonctionnalités efficace la création et l'édition de classes, de propriétés et d'instances, ainsi que des fonctionnalités d'inférence, de débogage et de tests d'inconsistance. Il prend le langage de modélisation OWL 2.0 Manchester DL que nous utilisons pour produire notre ontologie. Enfin, il intègre facilement nos autres éléments de développement à travers son menu d'extension [68].

Le deuxième élément est le raisonneur Pellet. À source ouverte et performant, il peut être utilisé pour inférer des connaissances à partir d'ontologies. Il prend en charge les normes OWL 2 et RDF, et s'intègre à Protégé. Son rôle est aussi de détecter les erreurs d'inconsistances de notre TBOX. Pellet raisonne par exemple qu'un avion militaire, inféré comme un avion commercial, serait inconsistant puisque que les deux classes sont disjointes. Le raisonneur permet ainsi à notre EEA d'offrir des inférences en toute logique.

Le troisième élément est Cellphie. Il s'agit d'un outil *open source* qui facilite la création de

l'ABOX d'*onto-by-wire*. À travers l'importation d'un tableau Excel contenant nos instances, nous pouvons, avec cet outil, produire des axiomes qui définissent comment les instances peupleront l'ontologie. La figure 5.7 présente des règles de transformation pour importer les axiomes des champs de messages ARINC 429 (*label*, *data*, SDI, SSM, parité). On importe pour un champ ses bits significatifs (*fbw : LSB* et *fbw : MSB*), sa valeur (*fbw : datafieldValue*), l'interprétation du champ (*fbw : decodedInformation*) et le message (ou paquet) ARINC 429 auquel appartient le champ (*fbw : isARINC429MessageOf*). Cellphie valide et sauvegarde aussi les règles que nous apportons à la ABOX [68].

```
1 Individual: @A*
2 Types: fbw:ARINC429Datafield
3 Facts: fbw:LSB @B*(xsd:integer),
4      fbw:MSB @C*(xsd:integer),
5      fbw:datafieldValue @D*(xsd:integer),
6      fbw:decodedInformation @E*(xsd:string),
7      fbw:isARINC429MessageOf @F*
```

FIGURE 5.7 Exemple de règles de transformation sous Cellphie.

Le quatrième élément est GraphDB. Il s'agit d'un système de gestion de bases de données orientées graphe compatible avec RDF et OWL. Dû à sa haute performance, nous produisons nos requêtes SPARQL pour répondre à nos questions de compétence avec cet outil plutôt que Protégé. Ses fonctionnalités de stockage favorisent l'analyse de relations d'*onto-by-wire*. C'est donc cet outil qui présente adéquatement aux parties prenantes l'information décortiquée par l'EEA. On y importe notre ontologie peuplée et les inférences raisonnées sous Protégé. Pour enrichir davantage notre modèle d'inférences, GraphDB est en plus configuré avec un ensemble de règles en *optimized* OWL2-QL (Query Language).

5.3.4 Analyse du modèle interdomaine d'*onto-by-wire*

L'identification de concepts et de leurs relations nous permet d'explorer la connectivité entre les différents domaines que nous intégrons dans notre ontologie. Ce que nous cherchons réellement à produire dans cette recherche est de relier la gestion du trafic aérien avec l'avionique et la cybersécurité pour expliquer des anomalies. Pour cela, il nous faut produire un modèle offrant l'interconnexion entre trois domaines : l'ARINC 429, l'aviation et la cybersécurité.

Analyse du modèle de l'ARINC 429

On obéit à la documentation de l'ARINC 429 pour identifier sa sémantique lors de la première itération sous la méthodologie SAMOD. Une approche du haut vers le bas est préconisé pour

cette itération puisque notre ontologie doit intégrer adéquatement le standard avionique. Seuls les avions commerciaux communiquent avec de l'ARINC 429 ou de l'AFDX. Les avions miliaires communiquent sous MIL-STD-1553.

La logique suit la décomposition d'un message ARINC 429 en plusieurs champs qui sont : le *label*, le *data*, la parité, le SDI et le SSM. On précise les trois types de messages ARINC 429 les plus courants : BNR, BCD et Discrete.

Puisque la signification de la valeur du SSM change selon le type de message, on prend en considération le type dans les règles d'équivalence. Les états de SSM possibles, sous-classes de *fbw : SSM*, sont : *normal*, *positive*, *negative*, *functional test*, *no computed data* et *failure warning* [27]. Par exemple, comme on le voit à la figure 5.8, pour les messages de types BNR, la sous-classe *failure warning* a pour règle d'équivalence la valeur 00 pour le SSM (*hasDatafieldValue xsd : integer*). Pour des données invalides autres qu'une erreur (*no computed data*) la valeur est 01. La valeur est 10 pour les tests fonctionnels souvent réalisés, en maintenance. Une valeur normale a enfin comme valeur 11. Notons que le status de code SSM change d'un type de message ARINC 429 à un autre. Il est pertinent de se fier au cahier d'exigences en cas de doute [27].

BNR data SSM Status Coding

BIT 31	BIT 30	Decoded Information
0	0	Failure Warning
0	1	No Computed Data
1	0	Functional Test
1	1	Normal Operation

FIGURE 5.8 Statut du code SSM des messages BNR [27].

Une représentation graphique de la TBOX d'*onto-by-429* est reproduite à la figure 5.9. On remarque les différents champs de données de l'ARINC 429 représentés sous forme de classes avec les différentes propriétés de celles-ci. Par exemple, le SDI est le champ qui lie un message ARINC 429 avec une composante avionique. Son rôle est en effet d'indiquer la source ou la destination d'un paquet.

On intègre aussi des exigences liées au nombre de LRU dans la communication simplex. Le transmetteur d'un LRU peut communiquer avec au plus 20 récepteurs. Chaque récepteur ne peut communiquer qu'avec un seul transmetteur de LRU. Pour continuer, le SDI précise le LRU dans la communication de deux avioniques. Si un message ARINC 429 détecté n'obéit pas aux règles, une inconsistance est retournée tout en précisant l'incohérence trouvée. Le tableau 5.3 illustre la sémantique de certains éléments significatifs pour adapter l'ARINC 429

à une ontologie.

Enfin, l'annexe C présente une part de la syntaxe de l'ontologie sous forme de logique descriptive fonctionnelle. On peut y lire les différentes règles d'équivalences et de sous-classes qui indiquent si la valeur d'un paquet ARINC 429 de type BCD est positive (plus, nord, est, droit) ou négative (moins, sud, ouest, gauche).

Analyse interdomaine d'ATMONTO et de l'ARINC 429

La seconde itération de développement de notre ontologie lie l'ARINC 429 avec ATMONTO. Pour offrir une ontologie mère (le *core*), on renomme l'ontologie *onto-by-wire* qui unie *onto-by-429* et ATMONTO. Une représentation graphique de la TBOX d'onto-by-wire est reproduite à la figure 5.10 dès sa deuxième itération. La syntaxe est ici plus axée sur l'interopérabilité des domaines. Certains termes, définis dans l'ARINC 429, sont reliés comme des sous-classes ou des classes équivalentes dans ATMONTO. Par exemple, la définition d'avion sous *onto-by-429* devient équivalente à celle de l'*equipment* d'ATMONTO. On intègre aussi des règles d'équivalence pour l'avionique dans les classes de système d'ingénierie aéronautique d'ATMONTO.

En parallèle, on préconise ici l'approche du bas vers le haut contrairement à la première itération. En collaborant avec un pilote pour cette recherche nous sommes à même apte à rédiger des questions de compétences pertinentes pour un vol critique. On intègre les classes de phases de vol et de condition climatique puisqu'elles sont des facteurs à considérer lors d'une accident. Les relations d'équivalence entre un message ARINC 429 et un LRU sont aussi réalisées en identifiant en premier lieu les communications nécessaires par un humain avant de concevoir l'avionique.

Pour simplifier notre contribution, nous ne présentons que les classes ou les propriétés produites dans cette recherche. Les zones *core* relient nos classes avec les deux ontologies qui ont été utilisées dans cette itération. Aussi, pour éviter de charger ce graphe, nous ne présentons que trois sous-classes sur huit de phase de vol (*flight phase*).

On rajoute aussi de nouvelles classes afin de faire le pont entre les messages ARINC et les communications. La classe *fwb : FlyByWireFix* est ajoutée dans le plan de vol pour préciser les messages ARINC 429 envoyés à une certaine période de temps. De plus, on intègre la phase de vol avec le *Trackpoint*. Il s'agit d'un détail essentiel dans la «défense en profondeur» puisque davantage d'accidents surviennent dans certaines phases de vol [69]. Tel qu'argumenté dans le chapitre précédent, une attaque informatique réussie sera vraisemblablement due à une accumulation de plusieurs facteurs (principe de «défense en profondeur»). Le tableau 5.4 illustre la sémantique d'éléments significatifs dans l'ontologie pour adapter *onto-by-429* avec ATMONTO.

Enfin, l'annexe C présente une part de l'ontologie sous forme de logique descriptive fonctionnelle. On peut y lire les différentes règles de propriétés et de sous-classes qui relient un point de piste (*atm : AircaftTrackPoint*) avec un message ARINC 429.

Analyse interdomaine de MITRE D3FEND, ATMONTO et onto-by-429

La corporation MITRE produit, avec MITRE D3FEND, une ontologie combinant le profil d'attaque, ses contre-mesures et les artéfacts ciblés tels des messages ARINC 429 ou des LRUs. Ses deux sous-classes mères sont d3fend : ATTACKThing pour l'aspect offensif et d3fend : D3FENDThing pour l'aspect défensif. Puisque le modus operandis du bord offensif et défensif sont intégré, on préconise une approche du bas vers le haut afin de bien saisir la volonté des parties ciblées. Pour y arriver, l'ontologie suit des scénarios d'attaques. On définit, d'une part, l'avionique et les commandes de vols électrique qui correspondent aux *d3fend : Artifact*. D'autre part, on adapte D3FEND pour lui offrir dans sa ABOX un scénario d'attaque composé d'une séquence d'identifiants de *techniques* (*d3fend : attack-id*) pour les séquences d'attaque.

Une représentation graphique de la TBOX d'onto-by-wire est reproduite à la figure 5.11. La syntaxe est ici plus axée sur la création de scénarios d'attaque soumis au cadre MITRE ATT&CK. Dans le graphe, on ajoute comme exemple l'identifiant d'attaque T1041. Il permet, lors de la production de nos requêtes SPARQL, de lier notre scénario à la classe d'une technique d'exfiltration par canaux. Certains termes, définis dans l'ARINC 429, sont reliés comme des sous-classes ou des classes équivalentes dans ATMONTO. Par exemple, la définition d'avion sous *onto-by-429* devient équivalente à celle de l'*equipment* d'ATMONTO. On intègre aussi des règles d'équivalence pour l'avionique dans les classes de système d'ingénierie aéronautique d'ATMONTO. Le tableau 5.5 illustre la sémantique d'éléments significatifs dans l'ontologie pour adapter D3FEND avec *onto-by-wire*.

Enfin, l'annexe C présente une part de la syntaxe de l'ontologie sous forme de logique descriptive fonctionnelle. On peut y lire les différentes règles de propriétés et de sous-classes qui relient une attaque offensive (*d3fend : next-ATTACK-scenario*) à une autre dans un même scénario d'attaque.

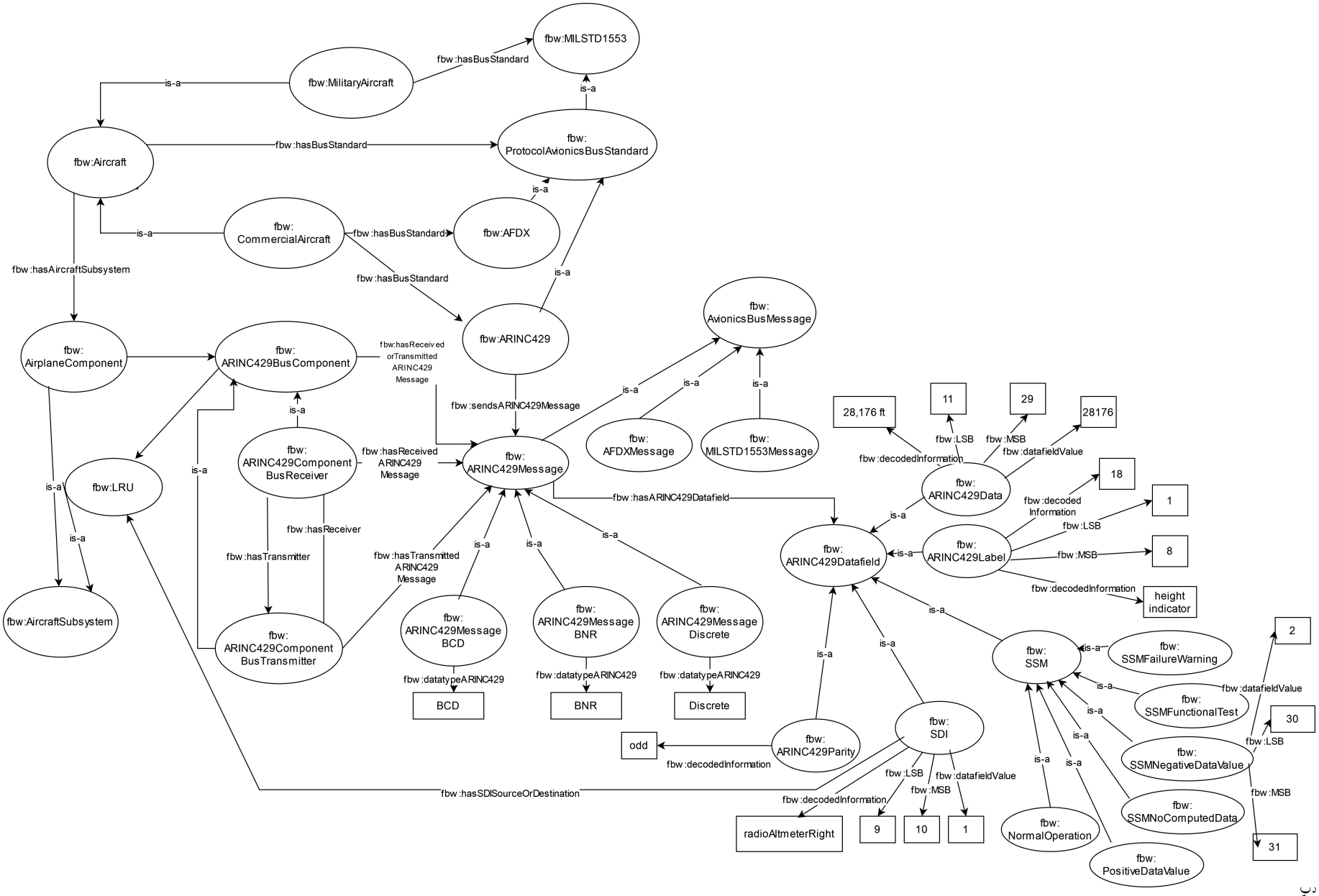


FIGURE 5.9 Représentation graphique de la TBOX d'onto-by-429

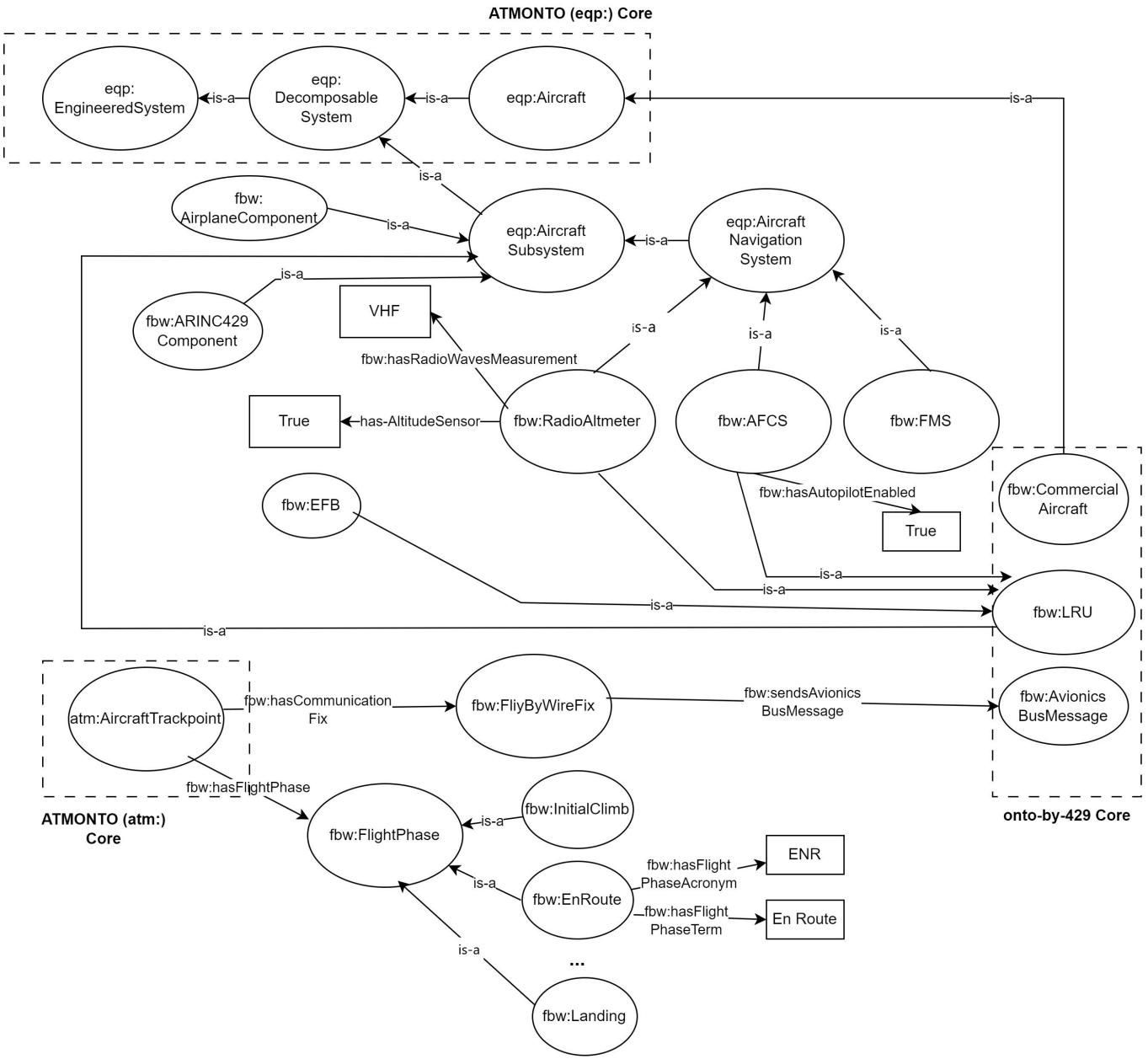


FIGURE 5.10 Représentation graphique de la TBOX d'ATMONTO liée à onto-by-wire

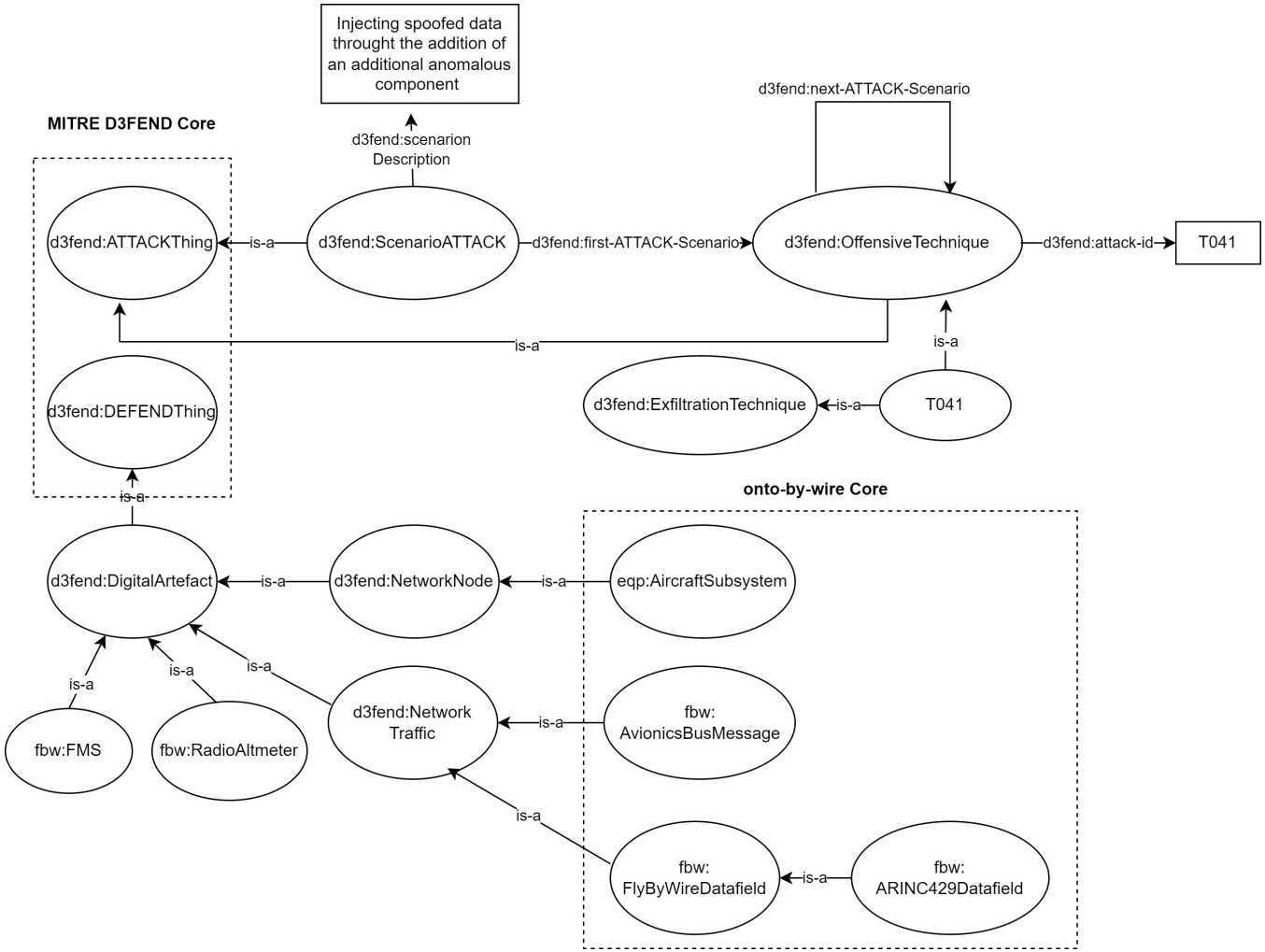


FIGURE 5.11 Représentation graphique de la TBOX de D3FEND liée à onto-by-wire

Instruction de la logique cognitive	Description de la sémantique	Modèle de donnée sémantique		
		Concept	Prédicat	Concept
L'avion communique avec des bus	L'avion a une composante	Aircraft	Has aircraft subsystem	Aircraft Component
	L'avion a un standard de bus	Aircraft	Has Bus Standard	Avionics bus standard
	L'avion commercial est un avion	Commercial Aircraft	Is a	Aircraft
	L'avion commercial a le standard ARINC429 ou AFDX	Commercial Aircraft	Has Bus Standard	ARINC 429, AFDX
L'avionique de l'avion est composée de LRU	La composante ARINC 429 communique avec de l'ARINC 429	Component ARINC 429	Has received or transmitted ARINC 429	ARINC 429 Message
	Le récepteur ARINC 429 possède au plus un transmetteur	Receiver ARINC 429	Has transmitter ARINC 429 < 2	Transmitter ARINC 429
	Le transmetteur ARINC 429 possède au plus 20 récepteurs	Transmitter ARINC 429	Has receiver ARINC 429 < 21	Receiver ARINC 429
	Le LRU a des composantes ARINC 429	LRU	Has ARINC 429 Bus Component	Component ARINC 429
Les messages ARINC 429 possèdent des types variés	Le message ARINC 429 est de type BCD, BNR ou Discrete	ARINC 429 Message	Is dataType	String
Le message ARINC 429 possède plusieurs champs de données	Un champ ARINC 429 a une position pour ses bits	Datafield ARINC 429	Bit position (LSB and MSB)	Integer
	Un champ ARINC 429 a une valeur pour sa donnée	Datafield ARINC 429	Datafield Value	Integer
	Data, Label, Parity, SDI, SSM sont des champs ARINC 429	Data, Label, Parity, SSM or SDI	Is a	Datafield ARINC 429
Le statuts d'un message ARINC 429 dépend de sa valeur et du type de message	Un message BCD "Normal" a un signe avec un SSM de 0 ou 3	Positive SSM ou Negative SSM	DatafieldValue == (0 or 3)	Integer
	Un message Discrete ou BNR "Normal"/"Failure" a un SSM de 0 ou 3	Normal SSM Failure SSM	DatafieldValue == (0 or 3)	Integer
	Un message sans données a un de SSM de 1	No computed data SSM	Datafield Value == 1	Integer
	Un message de test a un SSM de 2	Functional Test SSM	Datafield Value == 2	Integer
La source ou destination d'un message ARINC 429 indique un LRU	Un SDI indique un LRU	SDI	has source or destination	LRU

Tableau 5.3 Sémantique de la conception du standard de bus ARINC 429

Instruction de la logique cognitive	Description de la sémantique	Modèle de donnée sémantique		
		Concept	Prédicat	Concept
L'avionique est un sous-système de l'avion	La composante avionique est un sous-système de l'avion	Aircraft Component	Is a	Aircraft Subsystem
	Un LRU est un sous-système de l'avion	LRU	Is a	Aircraft Subsystem
Certains LRUs sont des systèmes de navigations	L'AFCS est un système de navigation	AFCS	Is a	Aircraft Navigation System
	L'AFCS a un autopilote	AFCS	Has autopilot enabled	Boolean
	Le radioaltimètre est un système de navigation	Radioaltmeter	Is a	Aircraft Navigation System
	Le radioaltimètre mesure des ondes radios et capte des altitudes	LRU	Has radiowaves and Has altitude sensors	Bolean
	Le FMS est un système de navigation	FMS	Is a	Aircraft Navigation System
Le plan de vol intègre les messages de bus	Le trackpoint a un fix de communication de bus	Trackpoint	Has Communication Fix	FlyByWire Fix
	Le fix de communication de bus envoie des messages de bus	FlyByWire FiX	Sends avionics bus message	Avionics Bus Message
Le plan de vol intègre les messages de bus	Le trackpoint a une phase de vol	Trackpoint	Has Flight Phase	Fligh Phase
	Il existe plusieurs phases de vol	Approach, En Route, Initial Climb, Take Off, etc..	Is a	Flight Phase
	La phase de vol a un acronyme et un terme	Fligh Phase	Has acronym and Has term	String

Tableau 5.4 Sémantique de la conception interdomaine du standard de bus ARINC 429 et de l'aviation.

Instruction de la logique cognitive	Description de la sémantique	Modèle de donnée sémantique		
		Concept	Prédicat	Concept
Un scénario est composé de techniques d'attaques	Le scénario fait partie de l'attaque	Attack Scenario	Is a	ATTACKThing
	Le scénario débute par une première technique d'attaque	Attack Scenario	First attack techniques	Offensive Technique
	Une technique d'attaque est suivi d'une technique d'attaque	Offensive Technique	Next attack techniques	Offensive Technique
L'avionique est un artefact	Le LRU est une ressource réseau	LRU	Is a	Network resource
	Le radioaltimètre est un artefact digital	Radioaltmeter	Is a	Digital Artifact

Tableau 5.5 Sémantique de la conception interdomaine, de la cybersécurité, du standard de bus ARINC 429 et de l'aviation.

5.3.5 Cas d'étude

Cette section présente les trois cas d'études auxquels nous avons choisi d'appliquer notre engin. Les cas obéissent au principe du fromage suisse de J. Reason [50]. On attribue la baisse de performance et les accidents à une accumulation d'événements imprévisibles qui incluent des erreurs humaines et des défaillances de systèmes. Nos scénarios appliquent ce principe en s'inspirant du célèbre cas du vol Air France Rio-Paris.

Le vol Air France 447, *en route* vers Paris, s'est écrasé dans l'océan Atlantique tuant tous ses passagers. La formation de cristaux de glace dans la sonde de pitot a causé une perte temporaire des indicateurs de vitesse. L'avion, un Airbus 330, survolait la tempête à travers des nuages avec un système anti-givrage correctement activé. C'est toutefois l'accumulation de facteurs qui auraient favorisé l'accident. En effet, durant ce vol, le capitaine n'ayant eu qu'une heure de sommeil la veille, est parti prendre une sieste et a laissé deux co-pilotes moins expérimentés responsables d'un contrôle complexe. Face à la perte des instruments de mesure, le pilote automatique a été désactivé. Dû à une confusion du fonctionnement du *fly-by-wire*, l'avion s'est mis en *stall*. Les co-pilotes se sont retrouvés inaptes à gérer la diminution de vitesse et le redressement de l'avion. En somme, ils ont plutôt augmenté le nombre d'erreurs humaines. L'accumulation d'alertes a ensuite contribué à augmenter leur charge cognitive. Malgré l'arrivée du capitaine au cockpit, ceux-ci n'ont pas réussi à redresser l'avion et équilibrer les vitesses. L'avion a continué sa descente pour s'écraser dans l'océan [70].

Une succession d'événements, anodins individuellement, rend une situation critique. Une attaque informatique peut devenir critique si l'on combine d'autres enjeux. Le principe de «défense en profondeur» est ainsi pris en compte. La figure 5.12 illustre les trois cas d'études qui sont appliqués à notre EEA en guise d'expérimentation.

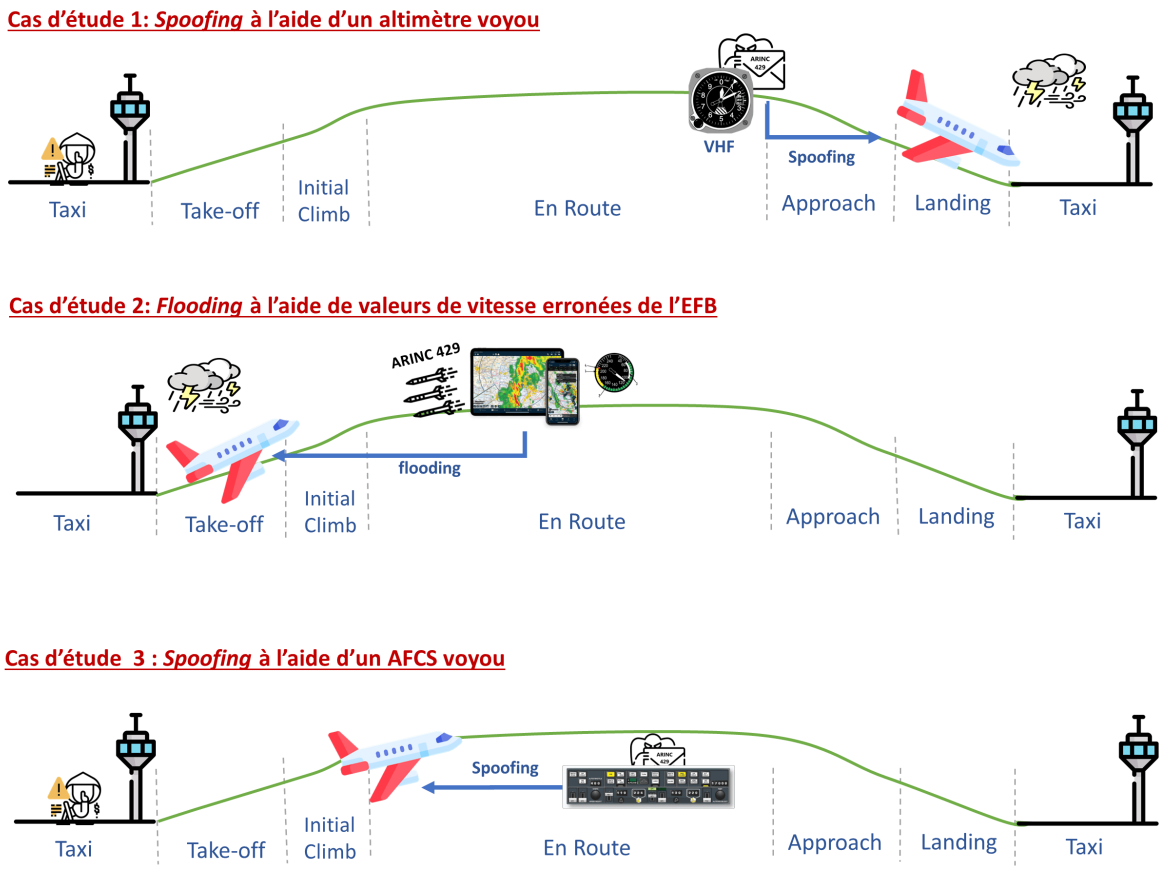


FIGURE 5.12 Représentation des trois cas d'études.

Cas d'étude 1

Le premier cas d'étude présente un avertissement lié à une attaque d'usurpation de paquet sur l'ACARS depuis un radioaltimètre corrompu qui vient affecter la trace d'un avion commercial, un Boeing 757, en direction de San Francisco vers l'aéroport KSFO. Le vol commence à New York depuis l'aéroport KJFK. Ce cas s'inspire du banc de test sous Triton [47] où une telle attaque vient interférer avec les opérations du système jusqu'au Communication Management Unit (CMU) et à l'ordinateur du FMS au point d'impacter la sûreté du vol. Puisque l'ACARS n'a aucune mesure d'authentification, nous n'avons en plus aucun moyen de déterminer si la source d'un message est légitime. Nous appliquons donc dans ce cas d'étude le scénario d'attaque de Spoonner (voir figure 5.2.2) qui est d'injecter des données usurpées à travers l'addition d'un composant additionnel voyou. Cette avionique envoie depuis une radio VHF trafiquée des paquets ARINC 429 vers le FMS qui affiche une altitude erronée au pilote.

Le SSM du paquet malicieux possède une trame standard avec une parité et une étiquette (*label*) valide ainsi qu'un SSM indiquant un statut *normal*. Spooner compromet ainsi la communication du bus ARINC 429.

Le cas est appliqué dans une situation où le pilote subit une forte charge de travail. Ce dernier est en phase d'atterrissage sous des conditions météo intenses, reporté par METAR, comprenant de la pluie, des orages et du brouillard avec très sûrement des turbulences dut au cisaillement du vent. La visibilité y est évaluée comme basse. Possédant un transpondeur de type S, un cas d'attaque de brouillage avec la tour de contrôle est aussi faisable [71]. Alors que l'altitude attendue à ce stade du vol devrait être 3600,0 pi, on reçoit une valeur nettement en dessous de 2583 pi.
Le message ARINC 429 est en compétition avec le message valide d'un autre récepteur qui affiche 3583 pi. Avec toutes ces conditions réunies, l'alerte est suffisamment intense pour en avertir les parties prenantes. Pour ce cas d'étude, le pilote en plein vol correspond à la partie prenante à laquelle nous souhaitons expliquer l'anomalie en cours. Les techniques de défense sont donc limitées aux actions directes d'isolement et d'éviction de l'attaque. On pose à l'EEA sa première question de compétence :

Est-ce que l'avion subit un cas d'attaque par spoofing où deux transmissions d'altitudes entrent en collision, et quelle est la mesure appropriée lors du vol ?

Titre de la requête : L'usurpation de l'altitude d'un avion lors de l'atterrissage
Définition de la requête : Deux valeurs d'altitudes provenant de paquets ARINC 429 rentrent en compétition. Les mauvaises conditions de vol rendent la situation propice à commettre une erreur humaine. Des contre-mesures sont proposées pour corriger l'avionique corrompue par une transmission malicieuse. La requête identifie la compétition présente lors de la transmission des paquets ARINC 429. Le tableau 5.6 liste les relations utilisées pour produire la requête et présente le domaine et la portée pour chacune des propriétés. La requête présentée à l'annexe B traduit la question de compétence en requête SPARQL.

Domain	Predicate	Range
ScenarioATTACK	scenarioDescription	xsd :string
ScenarioATTACK	first-ATTACK-scenario	OffensiveTechnique
OffensiveTechnique	next-ATTACK-scenario* / attack-id	xsd :string
DefensiveTechnique	d3fend-id	xsd :string
Flight	hasActualRoute	ActualFlightRoute
Flight	departureAirport	Airport
Flight	aircraftFlown	Aircraft
Aircraft	modeSCode	xsd :integer
ActualFlightRoute	hasFirstItem / hasNextItem*	AircraftTrackPoint
AircraftTrackPoint	hasFlightPhase	FlightPhase
AircraftTrackPoint	hasCommunicationFix / sendsFlyByWireMessage	AvionicsBusMessage
AircraftTrackPoint	aircraftFix / genAltitude	xsd :string
FlightPhase	flightPhaseAcronym **OR** flightPhaseTerm	xsd :string
ARINC429Message	hasARINC429Datafield	ARINC429Datafield
ARINC429Datafield	LSD **OR** MSB	xsd :integer
SDI	hasSDISourceORDestination / isLRUOf	LRU
Airport	hasMETARreport	METARReport
MeteorologicalCondition	hasWeatherCondition	WeatherCondition
MeteorologicalCondition	hasVisibilityCondition	VisibilityCondition
WeatherCondition	weatherIntensity	xsd :string
VisibilityCondition	unlimitedVisibility	xsd :boolean

Tableau 5.6 Vocabulaire de l'usurpation de l'altitude de l'avion lors de l'atterrissage.

Cas d'étude 2

Le deuxième cas d'étude présente un avertissement lié à une attaque de DOS sur le FMS avec comme porte d'entrée un Electronic Flight Bag (EFB) corrompu sur la chaîne d'approvisionnement qui vient affecter la disponibilité du service d'un avion commercial, toujours un Boeing 757, en direction de San Francisco vers l'aéroport KSFO. Le vol commence à New York depuis l'aéroport KJFK. Ce cas s'inspire du DOS lancé jusqu'aux connexions réseaux de l'AID pour affecter la négociation de trajectoire ciel à terre au point d'impacter la sûreté du vol [48]. Puisque l'EFB est corrompu depuis son approvisionnement, nous n'avons aucun moyen de déterminer si la source d'un message est légitime. La détection est laissée à l'analyse de la logique des paquets.

Nous appliquons donc dans ce cas d'étude le scénario d'attaque de Denise (voir Figure 5.2.2) qui est d'inonder des données à travers un LRU corrompu sur la chaîne d'approvisionnement *hardware*. L'avionique envoie depuis l'EFB des paquets ARINC 429 vers le FMS, puis l'anémomètre qui affiche une vitesse erronée au pilote. Le SSM du paquet malicieux possède une trame standard avec une parité et une étiquette (*label*) valide ainsi qu'un SSM indiquant un statut *normal*.

Denise compromet la communication du bus ARINC 429. Le cas est appliqué dans une situation où le pilote subit une forte charge de travail. Ce dernier est en phase de *take-off* sous, encore une fois des conditions météo intenses, comprenant de la pluie, des orages et du brouillard avec des turbulences causées par le cisaillement du vent. La visibilité y est évaluée faible. L'avion possède encore une fois un transpondeur de type S. Alors que la vitesse au sol attendue à ce stade du vol devrait être de 189 m/s, on reçoit une valeur nettement en dessous de 19 m/s.

Le message ARINC 429 est en incohérence avec le message attendu. Avec toutes ces conditions réunies, l'alerte devient suffisamment intense pour en avertir la maintenance qu'on choisit comme une partie prenante dans ce cas. Les techniques de défense sont donc limitées aux actions directes d'isolement et d'éviction de l'attaque. On pose à l'EEA sa deuxième question de compétence :

Est-ce que l'avion subit un cas d'attaque par déni de service où l'écart de valeur transmise est beaucoup trop élevé pour ce qui est attendu, et quelle est la mesure appropriée lors de la maintenance ?

Titre de la requête : Le *flooding* de l'indicateur de vitesse d'un avion lors du *take-off*
Définition de la requête : Les paquets reçus par l'ARINC 429 ont une valeur incohérente avec la valeur attendue. Les mauvaises conditions de vol rendent la situation propice à commettre une erreur humaine. Des contre-mesures sont proposées pour corriger la situation. La requête identifie l'écart de vitesse reçu par le paquet ARINC 429. Le tableau 5.7 liste les relations utilisées pour produire la requête et présente le *domain* et le *range* pour chacune des propriétés.

Domain	Predicate	Range
ScenarioATTACK	scenarioDescription	xsd :string
ScenarioATTACK	first-ATTACK-scenario	OffensiveTechnique
OffensiveTechnique	next-ATTACK-scenario* / attack-id	xsd :string
DefensiveTechnique	d3fend-id	xsd :string
Flight	hasActualRoute	ActualFlightRoute
Flight	departureAirport	Airport
Flight	aircraftFlown	Aircraft
Aircraft	modeSCode	xsd :integer
ActualFlightRoute	hasFirstItem / hasNextItem*	AircraftTrackPoint
AircraftTrackPoint	hasFlightPhase	FlightPhase
AircraftTrackPoint	hasCommunicationFix / sendsFlyByWireMessage	AvionicsBusMessage
AircraftTrackPoint	hasCommunicationFix / groundSpeed	xsd :integer
FlightPhase	flightPhaseAcronym **OR** flightPhaseTerm	xsd :string
ARINC429Message	hasARINC429Datafield	ARINC429Datafield
ARINC429Datafield	LSD **OR** MSB	xsd :integer
SDI	hasSDISourceORDestination / isLRUOf	LRU
Airport	hasMETARreport	METARReport
MeteorologicalCondition	hasWeatherCondition	WeatherCondition
MeteorologicalCondition	hasVisibilityCondition	VisibilityCondition
WeatherCondition	weatherIntensity	xsd :string
VisibilityCondition	unlimitedVisibility	xsd :boolean

Tableau 5.7 Vocabulaire de l'inondation de la vitesse au sol de l'avion lors du décollage.

Cas d'étude 3

Le troisième cas d'étude a comme particularité d'utiliser l'ICD d'une véritable simulation de vol avec de véritables paquets ARINC 429 formatés en paquet AFDX. Ces instances éliminent davantage le biais de l'expérimentateur que les paquets ARINC 429 produits dans les cas d'étude 1 et 2 qui obéissent quand même au standard ARINC 429 en ayant été toutefois produits par l'expert en ontologie.

Le cas présente un avertissement lié à une attaque d'usurpation de paquet sur le pilote automatique depuis un AFCS corrompu qui vient affecter la trace d'un avion commercial, un Bombardier Challenger 7500, lors d'un vol simulé de 16 h à 17 h. Ce cas s'inspire de l'expérience de Patrick Kiley qui simule la désactivation du pilote automatique d'un bus CAN [28]. Une telle attaque vient interférer avec les opérations du système jusqu'à l'*autothrottle* au point de cause d'une potentielle perte de contrôle de l'avion. Nous appliquons donc dans ce cas d'étude le scénario d'attaque de Spoonner (voir Figure 5.2.2) qui est d'injecter des données usurpées à travers l'addition d'un composant additionnel voyou. Cette avionique envoie depuis l'AFCS trafiquée des paquets ARINC 429 vers le PFD qui affiche une altitude erronée au pilote. Le SSM du paquet malicieux possède une trame standard avec une parité

et une étiquette (*label*) valide ainsi qu'un SSM indiquant un statut *normal*.

Spooner compromet ainsi la communication du bus ARINC 429. Puisque cette simulation n'incorpore pas de données météorologiques, nous omettons cette fois l'ajout de données METAR, toutefois nous argumentons que de mauvaises conditions météorologiques seraient suffisantes pour causer une surcharge de travail aux membres de l'équipage. L'avion enclenche sa phase de croisière *en route*. Alors que le pilote automatique devrait être enclenché à ce stade du vol, on reçoit une désactivation suspecte.

Puisqu'il s'agit d'une simulation de vol, on prend comme cible de l'attaque un développeur de simulateur de vol sécuritaire. On pose à l'EEA sa troisième question de compétence :

Est-ce que le vol simulé subit un cas d'attaque par spoofing où le pilote automatique est désactivé inhabituellement, et quelle est la contre-défense appropriée ?

Titre de la requête : L'usurpation du pilote automatique d'un avion en croisière
Définition de la requête : Une désactivation inhabituelle du pilote automatique durant la phase de croisière survient. Des contre-mesures sont proposées pour corriger l'avionique corrompue par une transmission malicieuse. La requête identifie l'avionique corrompue lors de la transmission des paquets ARINC 429. Le tableau 5.8 liste les relations utilisées pour produire la requête et présente le *domain* et le *range* pour chacune des propriétés.

Domain	Predicate	Range
ScenarioATTACK	scenarioDescription	xsd :string
ScenarioATTACK	first-ATTACK-scenario	OffensiveTechnique
OffensiveTechnique	next-ATTACK-scenario* / attack-id	xsd :string
DefensiveTechnique	d3fend-id	xsd :string
Flight	hasActualRoute	ActualFlightRoute
Flight	departureAirport	Airport
ActualFlightRoute	hasFirstItem / hasNextItem*	AircraftTrackPoint
AircraftTrackPoint	hasFlightPhase	FlightPhase
AircraftTrackPoint	hasCommunicationFix / sendsFlyByWireMessage	AvionicsBusMessage
FlightPhase	flightPhaseAcronym **OR** flightPhaseTerm	xsd :string
AFCS	hasAutoPilotEnable	xsd :boolean
ARINC429Message	hasARINC429Datafield	ARINC429Datafield
ARINC429Datafield	LSD **OR** MSB	xsd :integer
SDI	hasSDISourceORDestination / isLRUOf	LRU
SSM	decodedInformation	xsd :string

Tableau 5.8 Vocabulaire de l'usurpation de l'altitude de l'avion lors de l'atterrissage.

5.3.6 Poids des contre-défenses

Pour déterminer les contre-défenses à appliquer, nous construisons sur SPARQL (avec CONSTRUCT) des graphes RDF qui indiquent le nombre d'occurrences pour chacune des contre-défenses envers les *techniques* MITRE ATT&CK du scénario d'attaque choisit pour le cas d'étude. Les occurrences au-dessus d'un certain seuil (produit avec COUNT) sont importées comme des contre-défenses potables lorsque nous posons les questions de compétences à notre EEA. Plus la valeur est grande, plus la contre-défense est jugée pertinente. Les triplets produits ont la sémantique :
d3fend : DefensiveTechnique rdfs :numberOfCounterMeasures xsd :integer
Le *listing* de l'annexe A présente le code de la requête utilisée pour construire le graphe RDF de poids de la contre-défense. Le triplet est ensuite importé dans la base de données SPARQL. Ces données aident dans la réponse de nos questions de compétence à filtrer, dans la colonne «contre-mesures», celles qui sont pertinentes à appliquer.

CHAPITRE 6 RÉSULTATS

Ce chapitre présente les résultats obtenus durant les travaux de ce mémoire. On présente les réponses aux questions de compétences de nos cas d'études avec la ABOX d'*onto-by-wire* peuplée en plus des tests d'inconsistances de la TBOX réalisée.

L'application de la méthodologie SAMOD nous a permis de produire l'ontologie *onto-by-wire* de notre EEA. Avec l'importation des ontologies ATMONTO et MITRE D3FEND, *onto-by-wire* compte au total 1667 classes, 340 propriétés d'objets et 247 propriétés de données.

On s'inspire de la littérature [12] pour produire les tableaux 6.1 et 6.2 qui présentent les résultats des axiomes de la TBOX de l'ontologie et de la ABOX pour les étapes du premier scénario d'attaque. On ne présente que les axiomes ayant été produits pour les expériences de ce mémoire et non les axiomes importés.

Étape	Phrase du modèle cognitif	Syntaxe de Protégé
Type de Bus	CommercialAircraft ***is-a*** Aircraft and ***has*** BusStandard (ARINC429 or AFDX)	CommercialAircraft ***SubClassOf*** Aircraft **and** hasBusStandard **some** (AFDX **or** ARINC429))
	ARINC429 ***is-a*** ProtocolAvionicsBusStandard and ***sends*** ARINC429Message	ARINC429 ***SubClassOf*** ProtocolAvionicsBusStandard **and** sendsARINC429Message **some** ARINC429Message
	ARINC429MessageBNR ***is-a*** ARINC429Message and ***has*** dataTypeARINC429 of BNR	ARINC429MessageBNR ***SubClassOf*** ARINC429Message **and** dataTypeARINC429 **value** "BNR"
Champs du Bus	ARINC429MessageBNR ***has*** ARINC429Datafield	ARINC429MessageBNR hasARINC429Datafield **some** ARINC429Datafield
	Label **is-a** ARINC429Datafield and ***has*** LSB of 1 or ***has*** MSB of 8	ARINC429Label ***SubClassOf*** ARINC429Datafield **and** (LSB **value** 1) **or** (MSB **value** 8)
	Data **is-a** ARINC429Datafield and ***has*** LSB of 9 or ***has*** MSB of 10	ARINC429Label ***SubClassOf*** ARINC429Datafield **and**(LSB **value** 9) **or** (MSB **value** 10)
	SDI **is-a** ARINC429Datafield and ***has*** LSB of 11 or ***has*** MSB of 19	ARINC429SDI ***SubClassOf*** ARINC429Datafield **and**(LSB **value** 11) **or** (MSB **value** 19)
LRU corrompu	SDI ***has*** SDISourceOrDestination ARINC429BusComponent	ARINC429SDI hasSDISourceOrDestination **some** ARINC429BusComponent
	ARINC429Transmitter ***is-a*** ARINC429BusComponent or ***has*** Transmitted ARINC429Message or **is** LRU of LRU	ARINC429ComponentBusTransmitter ***SubClassOf*** AirplaneComponent hasTransmittedARINC429Message **some** ARINC429Message isLRUOf **some** LRU
Phase de vol	Landing ***is-a*** FlightPhase and ***has*** acronym LDG or ***has*** term Landing	Landing ***SubClassOf*** FightPhase **and** (flightPhaseAcronym **value** LDG) **or** (flightPhaseTerm **value** Landing)
Scenario d'attaque	ScenarioATTACK ***is-a*** ATTACKThing and ***has*** scenarioDescription	ScenarioATTACK ***SubClassOf*** ATTACKThing **and** scenarioDescription **some** *xsd :string*
	ScenarioATTACK ***has*** first-ATTACK-scenario OffensiveTechnique	first-ATTACK-scenario ***Domains*** : ScenarioATTACK ***Ranges*** : OffensiveTechnique
	OffensiveTechnique ***has*** next-ATTACK-scenario OffensiveTechnique	next-ATTACK-scenario ***Domains*** : OffensiveTechnique ***Ranges*** : OffensiveTechnique

Tableau 6.1 Axiomes de la TBOX.

Étape	Phrase du modèle cognitif	Syntax de Protégé
Type de Bus	AircraftN713TW ***is-instance-of*** CommercialAircraft	Aircraft N713TW ***Types*** CommercialAircraft
	AircraftN713TW ***has*** BusStandard aRINC429	AircraftN713TW ***hasBusStandard*** aRINC429
	aRINC429 ***is-instance-of*** ARINC429	aRINC429 ***Types*** ARINC429
	aRINC429 ***sends*** ARINC429Message7	aRINC429 ***sendsARINC429Message*** ARINC429Message7
	ARINC429Message7 ***is-instance-of*** ARINC429MessageBNR	ARINC429Messag7 ***Types*** ARINC429MessageBNR
Champs du Bus	ARINC429Message7 ***has***ARINC429Datafield label6	ARINC429Message7 ***hasARINC429Datafield*** label7
	label7 ***is-instance-of*** Label	label7 ***Types*** ARINC429Label
	label7 ***has***label 370	label7 ***hasDatafieldValue*** 370
	data7 ***is-instance-of*** Data	data7 ***Types*** ARINC429Data
	data7 ***has***data 2583	data7 ***hasDatafieldValue*** 2583
	sdi7 ***is-instance-of*** SDI	sdi7 ***Types*** ARINC429SDI
LRU corrompu	sdi7 ***has***Source RAt1	sdi7 ***hasSDISourceOrDestination*** RAt1
	RAt1 ***is-instance-of*** ARINC429Transmitter	RAt1 ***Types*** ARINC429ComponentBusTransmitter
	RAt1 **is**LRU of RA	RAt1 ***isLRUOf*** RA
	RA ***has***RadioWavesMeasurement VHF	RA ***hasRadioWavesMeasurement*** VHF
	RA ***has***AltitudeSensor true	RA ***hasAltitudeSensor*** true
	RA ***is-instance-of*** RadioAltmeter	RA ***Types*** RadioAltmeter
Phase de vol	landing ***is-instance-of*** FlightPhase	landing ***Types*** FlightPhase
	landing ***is-instance-of*** Landing	landing ***Types*** Landing
Scenario d'attaque	ScenarioATTACK1 ***is-instance-of*** ScenarioATTACK	ScenarioATTACK1 ***Types*** ScenarioATTACK
	ScenarioATTACK1 ***has***first-ATTACK T1200-1	ScenarioATTACK1 **first-ATTACK-scenario** T1200-1
	T1200-1 ***is-instance-of*** OffensiveTechnique	T1200-1 **Types** OffensiveTechnique
	T1200-1 ***has***next-ATTACK T1204.002-1	T1200-1 ***next-ATTACK-scenario*** T1204.002-1
	T1204.002-1 ***is-instance-of*** OffensiveTechnique	T1204.002-1**Types** OffensiveTechnique

Tableau 6.2 Axiomes de la ABOX.

6.1 Réponse aux questions de compétence

La validation de la logique de la ABOX sous GraphDB montre que l'Engin d'Explication d'Anomalies répond à des questions de compétence qui ne prennent pas en compte seulement le bas niveau du *fly-by-wire*, mais aussi la logique des paquets. Quant au raisonneur Pellet, il permet à l'ontologie de tirer des inférences pour offrir des réponses attendues. Nos résultats montrent ainsi que nous pouvons implémenter des requêtes établissant des relations sémantiques avec trois domaines : l'ARINC 429, l'aviation et la cyberdéfense.

La base de connaissances utilisent les instances peuplant l'ontologie qui sont des jeux de données instanciées et inférées avec Pellet dans la ABOX. La base de connaissances intègre ainsi conjointement la base de données produite avec l'ICD de Bombardier et celle de la NASA. On compte, selon Protégé 151 instances, 90 assertions de classe, et 477 assertions de propriétés. Les divers cas de test réalisés assurent que l'EEA a été apte à modéliser la connaissance.

Après avoir importé notre base de connaissances contenant nos questions de compétences et lancé le raisonneur Pellet, l'ontologie raisonne des inférences. Elles sont ensuite exportées sous GraphDB. Pour *onto-by-429*, GraphDB dénombre, à l'importation, 2,006 déclarations logiques et 5,782 déclarations supplémentaires inférées pour un total de 7,788 déclarations.

Trois cas d'études sont aussi réalisés sous SPARQL avec les instances produites par l'on-

tologie. Un graphe RDF contenant les scénarios d'attaque A ou B avec les identifiants des *techniques* du MITRE a été importé pour chacun des cas. Les requêtes des trois scénarios sont présentées à l'annexe B.

6.1.1 Réponse à la question de compétence du cas 1

Est-ce que l'avion subit un cas d'attaque par usurpation de paquet (spoofing) où deux transmissions d'altitudes entrent en collision, et quelle est la mesure appropriée lors du vol ?

L'EEA indique dans le cas d'étude 1 au tableau 6.3 que deux altitudes entrent bien en collision. On présente le point de piste (*trackpoint*), une description de l'attaque, son vecteur, l'avionique compromis, les différences d'altitudes ainsi que la contremesure souhaitée. On propose au pilote de filtrer certaines parties du réseau. Ici, contrairement au radioaltimètre, le pilote peut se fier à d'autres instruments avioniques comme le GPS ou un altimètre redondant qu'il sait fiables pour mesure l'altitude.

Attack Index	Warning	Timestamp	Attack Scenario	Attack Vector	Compromised Device	Erroneous Parameter	Data Received	Data Expected	Counter Measure	Counter Measure Detail
1	Warning : More than one altitude value	TrackPoint for AAL35 ON 2014-07-15 at 18 :48	Injecting spoofed data through the addition of an additional anomalous component	Compromised ARINC 429 Bus	Radio Altimeter	Decision Height	3582	3600	Isolate : Network Traffic Filtering	Restricting network traffic originating from any location.
2	Warning : More than one altitude value	TrackPoint for AAL35 ON 2014-07-15 at 18 :48	Injecting spoofed data through the addition of an additional anomalous component	Compromised ARINC 429 Bus	Radio Altimeter	Decision Height	2582	3600	Isolate : Network Traffic Filtering	Restricting network traffic originating from any location.

Tableau 6.3 Résultat du traçage de l'attaque du cas d'étude 1.

6.2 Réponse à la question de compétence du cas 2

Est-ce que l'avion subit un cas d'attaque de déni de service par inondation de paquets (flooding) où l'écart de valeurs transmises est beaucoup trop élevé pour ce qui est attendu, et quelle est la mesure appropriée lors de la maintenance ?

L'EEA indique dans le cas d'étude 2 au tableau 6.4 que l'incohérence de paquets avioniques est détectée. On présente le point de piste, une description de l'attaque, son vecteur, l'avionique compromise, les différences d'altitudes ainsi que la contre-mesure souhaitée. Les contre-mesures les plus adaptées pour la maintenance dans notre cas consistent à déployer un leurre dans le réseau pour limiter la portée de l'EFB. Une autre mesure à faire en parallèle serait de réaliser une évaluation de l'inventaire de l'avionique et l'architecture réseau de l'avion afin de détecter les vulnérabilités potentielles et les vecteurs d'attaques.

Attack Index	Warning	Timestamp	Attack Scenario	Attack Vector	Compromised Device	Erroneous Parameter	Data Received	Data Expected	Counter Measure	Counter Measure Detail
1	Warning : Anomalous Ground Speed	TrackPoint for AAL35 ON 2014-07-15 at 17 :52	Initiating a denial of service through a malicious update perpetrated by an attack on the hardware chain of supplies	Compromised ARINC 429 Bus	Electronic Flight Bag	Ground Speed	19	189	Model : D3-AVE Asset Vulnerability Enumeration	Asset vulnerability enumeration enriches inventory items with knowledge identifying their vulnerabilities.
2	Warning : Anomalous Ground Speed	TrackPoint for AAL35 ON 2014-07-15 at 17 :52	Initiating a denial of service through a malicious update perpetrated by an attack on the hardware chain of supplies	Compromised ARINC 429 Bus	Electronic Flight Bag	Ground Speed	19	189	Deceive : D3-DNR Decoy Network Resource	Deploying a network resource for the purposes of deceiving an adversary.

Tableau 6.4 Résultat du traçage de l'attaque du cas d'étude 2.

6.3 Réponse à la question de compétence du cas 3

Est-ce que le vol simulé subit un cas d'attaque par spoofing où le pilote automatique est désactivé inhabituellement et quelle est la contre-défense appropriée ?

L'EEA indique dans le cas d'étude 3 au tableau 6.5 que l'incohérence de paquets avioniques est détectée. On présente le point de piste, une description de l'attaque, son vecteur, l'avionique compromis, le paramètre erroné ainsi que la contre-mesure souhaitée. On ne montre que les contre-mesures adaptées pour un développeur d'IDS se focalisant sur la tactique de *détection*. Les techniques recommandées par notre EEA sont : un profilage de la charge utile entre le client et le serveur, une déviation vers un trafic réseau communautaire, une analyse du ratio *download-upload* de chaque hôte, une détection de métadonnées anormales, une détection sur la présence de sessions à distance et enfin une analyse des données de localisations. Chacune de ces techniques de MITRE D3FEND possède un identifiant ainsi qu'une brève description.

Attack Index	Warning	Timestamp	Attack Scenario	Attack Vector	Compromised Device	Erroneous Parameter	Data Received	Counter Measure	Counter Measure Detail
1	Warning : Anomalous Auto Pilot Disabled	TrackPoint for ESIM0809 at 2020-09-1516 :36 :30	Injecting spoofed data through the addition of an additional anomalous component	Compromised ARINC 429 Bus	Left Automatic Flight Control System	Autopilot Status	AP OFF	Detect : D3-CSPP Client-server Payload Profiling	Comparing client-server request and response payloads to a baseline profile to identify outliers
2	Warning : Anomalous Auto Pilot Disabled	TrackPoint for ESIM0809 at 2020-09-1516 :36 :30	Injecting spoofed data through the addition of an additional anomalous component	Compromised ARINC 429 Bus	Left Automatic Flight ControlSystem	Autopilot Status	AP OFF	Detect : D3-NTCD Network Traffic Community Deviation	Establishing baseline communities of network hosts and identifying statistically divergent intercommunity communication.
3	Warning : Anomalous Auto Pilot Disabled	TrackPoint for ESIM0809 at 2020-09-1516 :36 :30	Injecting spoofed data through the addition of an additional anomalous component	Compromised ARINC 429 Bus	Left Automatic Flight Control System	Autopilot Status	AP OFF	Detect : D3-PHDURA Per Host Download-Upload Ratio Analysis	Detecting anomalies that indicate malicious activity by comparing the number of data downloaded versus data uploaded by a host.
4	Warning : Anomalous Auto Pilot Disabled	TrackPoint for ESIM0809 at 2020-09-1516 :36 :30	Injecting spoofed data through the addition of an additional anomalous component	Compromised ARINC 429 Bus	Left Automatic Flight ControlSystem	Autopilot Status	AP OFF	Detect : D3-PMAD Protocol Metadata Anomaly Detection"	Collecting network communication protocol metadata and identifying statistical outliers."
5	Warning : Anomalous Auto Pilot Disabled	TrackPoint for ESIM0809 at 2020-09-1516 :36 :30	Injecting spoofed data through the addition of an additional anomalous component	Compromised ARINC 429 Bus	Left Automatic Flight Control System	Autopilot Status	AP OFF	Detect : D3-RTSD Remote Terminal Session Detection"	Detection of an unauthorized live remote terminal console session by examining network traffic to a network host."
6	Warning : Anomalous Auto Pilot Disabled	TrackPoint for ESIM0809 at 2020-09-1516 :36 :30	Injecting spoofed data through the addition of an additional anomalous component	Compromised ARINC 429 Bus	Left Automatic Flight ControlSystem	Autopilot Status	AP OFF	Detect : D3-UGLPA User Geolocation Logon Pattern Analysis"	Monitoring geolocation data of user logon attempts and comparing it to a baseline user behaviour profile to identify anomalies in logon location

Tableau 6.5 Résultat du traçage de l'attaque du cas d'étude 3.

6.4 Tests d'inconsistance de la TBOX

Nous vérifions la logique de la TBOX d'*onto-by-wire* en posant des tests d'inconsistance tout au long des itérations. On ajoute volontairement, pour les tests, des instances qui n'obéissent pas à notre ontologie ou à l'inverse qui devraient retourner davantage d'inférences. On compte au total 17 tests. Les tests de consistance de la première itération s'assurent que le comportement d'un paquet ARINC 429 est adéquat. Par exemple, si un récepteur possède deux transmetteurs, une erreur logique est retournée. À la seconde itération, on vérifie plutôt l'interopérabilité entre ATMONTO et *onto-by-429*. Un paquet ARINC 429 doit pouvoir inférer son point de piste (atm : AircraftTrackpoint). Pour la troisième itération, on vérifie que nos techniques offensives infèrent leur scénario avec notre nouvelle classe, *d3fend : ATTACKScenario*.

Le raisonneur Pellet réussit à repérer toutes les incohérence des inférences et des équivalences pour chaque nouvelle classe créée. Le tableau 6.6 présente les tests d'inconsistances réalisés durant de chaque itération.

Test d'inconsistance	Description	Résultat
Test 1.1	L'ontologie génère des avions capables d'inférer s'ils sont commerciaux ou militaires selon le type de bus (AFDX, MIL-STD-1553, ARINC429)	***Infered*** : CommercialAircrafta (Aircraft **and** hasBusStandard AFDX)
Test 1.2	L'ontologie qui génère un avion commercial usant un bus militaire MIL-STD-1553 affiche une erreur d'inconsistance	***owl :Nothing*** : MilitaryAircraft (Aircraft **and** hasBusStandard ARINC429)
Test 1.3	L'ontologie génère un champ de données ARINC429 qui réussit à inférer son type de message (BCD, BNR, Discrete)	***Infered*** : ARINC429MessageDiscrete (ARINC429Message **and** dataTypeARINC429 "Discrete")
Test 1.4	L'ontologie génère une inconsistance en cas de bit significatif erroné	***owl :Nothing*** : ARINC429Label (ARINC429Datafield **and** MSB value 7)
Test 1.5	L'ontologie génère une inconsistance si la valeur d'un champ est trop grande	***owl :Nothing*** : ARINC429Data (ARINC429Datafield **and** datafieldValue 163841481)
Test 1.6	L'ontologie génère une inconsistance si le type de valeur, par exemple un nombre entier (integer) n'est pas respecté	***owl :Nothing*** : ARINC429Data (ARINC429Datafield **and** datafieldValue 0.2)
Test 1.7	L'ontologie infère si un LRU transfert ou reçoit un message ARINC 429	***Infered*** : ARINC429ComponentBusReceiver hasReceiver **Range** ARINC429ComponentBusReceiver
Test 1.8	L'ontologie génère une inconsistance, si une source se transfert à elle-même (propriété Irréflexive et Asymétrique)	***owl :Nothing*** : (*Irreflexive* or *Asymmetric*) hasTransmitter transmitter1 hasTransmitter transmitter1
Test 1.9	L'ontologie génère une inconsistance s'il y a plus d'un transmetteur pour un capteur	***owl :Nothing*** : (*Cardinality 1*) hasTransmitter receiver1 hasTransmitter transmitter 1 **and** receiver1 hasTransmitter transmitter 2
Test 2.1	Une phase de vol avec un acronyme invalide génère une inconsistance.	***owl :Nothing*** : Approach Approach flightPhaseTerm "Landing" **and** Approach flightPhaseAcronym "LDG"
Test 2.2	Une phase de vol peut inférer sa phase de vol avec l'acronyme ou le terme de vol	***Infered*** : Landing Landing flightPhaseTerm "Landing" **and** Approach flightPhaseAcronym "LDG"
Test 2.3	Un avion eqp :Airplane est aussi un avion fbw :Airplane	***Inferred*** : eqp :Aircraft *eqp* :Aicraft ***EquivalentTo*** *fbw* :Airplane
Test 2.4	L'attribut fbw :hasCommunication infère qu'un *trackpoint* possède un fix de communication (fbw :FlyByWireFix)	***Inferred*** : AircraftTrackPoint **or** FlyByWireFix hasCommunication **Domain** AircraftTrackPoint and hasCommunication **Range** FlyByWireFix
Test 2.5	Un message MIL-STD-1553 ne peut être le fix de communication d'un paquetARINC429	***owl :Nothing*** : FlyByWireFix sendsARINC429Messge **Range** MIL-STD-1553
Test 2.6	L'autopilote ne peut posséder qu'un état à la fois	***owl :Nothing*** : (*Functional*) **Range** hasAutoPilotEnabled hasAutoPilotEnabled true **and** hasAutoPilotEnable false
Test 3,1	L'ontologie infère un scénario d'attaque	***Inferred*** : ScenarioATTACK ATTACKThing **and** scenarioDescription xsd :String
Test 3,2	L'ontologie infère un EFB	***Infered*** : EFB LRU **and** IsEFB xsd :Boolean

Tableau 6.6 Résultats des tests d'inconsistance produits au fil de chaque itération.

6.5 Tests de compétence de la ABOX

Lors de chacune des itérations, nous avons validé les *modelets* en posant à l'ontologie des questions de compétence. Ces tests sont fait à l'aide de requêtes SPARQL. Lorsque les réponses à ces questions sont cohérentes on peut poursuivre l'implémentation de l'ontologie. L'utilité de nos tests permet également de valider la variété de questions qui peuvent ensuite être posées. Nous validons par exemple à l'itération 1, uniquement l'ontologie *onto-by-429*. Ces question sont intégrées ensuite à nos trois cas d'études finaux pour décrire l'avionique comportant des anomalies. Pour l'itération 2, nous validons le comportement interdisciplinaire entre *onto-by-429* et ATMONTO.

Le tableau 6.7 présente, comme pour les réponses de nos trois cas d'étude, certaines questions ayant été adressées à *onto-by-wire* et ses réponses validées. Il se peut aussi qu'une question de compétence renvoie plus d'une réponse.

Le tableau 6.8 énumère les tests de compétences intermédiaires en plus des trois cas d'études posés à notre ontologie à la troisième itération. La question du test 1.2, par exemple, demande les sources et destinations, obtenues du SDI, à un paquet ARINC 429. La requête revoit le nom du paquet *ARINC429Message1*, la source ou la destination, étant un récepteur de l'ACARS, et une description du paquet qui décrit que la source de ce paquet ARINC 429 provient du radioaltimètre.

Le nombre de questions posées est de 12. Les questions 1.1 et 1.5 n'ont pas été réutilisées dans la création de nos cas d'études. Une quatrième question de compétence aurait pu tenter de prendre en compte le cas d'un attaquant qui rajoute un second transmetteur lorsque le premier est inactif avec ces requêtes [6]. Ayant toutefois limité notre dernière itération à trois questions de compétences, notre méthodologie nous a obligée à omettre ce cas. Pour ce qui est des questions 2.x, elles correspondent à une ébauche de nos trois cas d'études 3.x sans l'intégration de MITRE D3FEND.

	Question de compétence	**Résultat**	**Type**
Test 1.1	Quelles composantes ARINC reçoit un message ARINC429 depuis plus d'un transmetteur ?	*FMSr1*	Un capteur avionique
Test 1.3.	La hauteur de décision indiquée par le label 370 est-elle cohérente avec une hauteur en phase de vol croisière ?	*Yes*	ASK Valeur booléenne
Test 2.1	Le ground speed d'un paquet ARINC 429 est-il cohérent avec la vitesse attendue ?	*ARINC429Message5* *Ground Speed* *189* *19*	Paquet ARINC 429 Caractéristique du paquet Vitesse attendue Vitesse reçue

Tableau 6.7 Réponses aux requêtes des questions de compétence intermédiaires.

	Question de compétences
Test 1.1	Quelles composantes ARINC reçoit un message ARINC429 depuis plus d'un transmetteur ?
Test 1.2	Quelles sont les sources et destinations des paquets ARINC 429 ?
Test 1.3	La hauteur de décision indiquée par le label 370 est-elle cohérente avec une hauteur en phase de vol croisière ?
Test 1.4	Quels LRUs envoie des paquets avec un identifiant valide (377) ?
Test 1.5	Combien de messages ARINC 429ont été reçus et transférés par le FMS ?
Test 1.6	Quels sont les statuts des messages de type BCD et BNR ?
Test 2.1	Le ground speed d'un paquet ARINC 429 est-il cohérent avec la vitesse attendue ?
Test 2.2	Deux altitudes en conflits sont-elles en phase de croisière détectées ?
Test 2.3	En phase de décollage sous de mauvaise condition métorologique, l'autopilote est-il activé ?
Test 3.1	Est-ce que l'avion subit un cas d'attaque par usurpation de paquet où deux transmissions d'altitudes entrent en collision, et quelle est la mesure appropriée lors du vol ?
Test 3.2	Est-ce que l'avion subit un cas d'attaque de déni de service par inondation de paquets où l'écart de valeurs transmises est beaucoup trop élevé pour ce qui est attendu, et quelle est la mesure appropriée lors de la maintenance ?
Test 3.3	Est-ce que le vol simulé subit un cas d'attaque par où le pilote automatique est désactivé inhabituellement et quelle est la contre-défense appropriée ?

Tableau 6.8 Liste des questions de compétences posées à l'ontologie

6.5.1 Comparaison entre notre EEA et la littérature

L'état de l'art nous a fait part de deux constats. La première mentionne l'importance d'informer aux parties prenantes de la présence de cyberattaque, le tout de façon standardisé. Les tests d'inconsistances nous ont montré que notre ontologie est apte à produire un langage qui demeure cohérent afin qu'il puisse être standardisé dans l'industrie. Son interopérabilité est démontrée grâce aux questions de compétences traitant d'un domaine interdisciplinaire posé à notre base de connaissances.

Le deuxième constat dont il était mention dans l'état de l'art a été la nécessité de développer davantage de technologies qui s'attardent à la sécurité logique des paquets pour accompagner les technologies de défenses basées sur les caractéristiques matérielles. Grâce aux capacités d'inférence de l'ontologie, nous avons développer un système pouvant expliquer des anomalies ou des comportements normaux avec uniquement des règles de logiques fonctionnelles.

Vu la carence de l'état de l'art face à ces deux enjeux, le nombre d'approches auxquelles nous pouvons comparer notre ontologie est limité. On retrouve néanmoins, l'engin d'explication d'anomalies utilisé par AnoMili face aux attaques d'usurpation de paquets [15]. Ce dernier a opté pour une modélisation arbitraire de ses résultats qui n'obéit à aucun standard et aucun processus d'inférence. On argumente qu'intégrer notre ontologie à AnoMili ou un autre type d'IDS offrirait ainsi une réponse face aux deux enjeux de la littérature que nous avons adressé.

CHAPITRE 7 CONCLUSION

7.1 Synthèse des travaux

Les attaques informatiques sont devenues une inquiétude préoccupante pour la sécurité de l'aviation. La rapide sophistication des aéronefs a produit des communications plus efficaces, au prix d'entraîner un écart considérable avec le domaine de la sécurité. Inspiré du principe de «défense en profondeur», ce mémoire a ciblé comme vecteur d'attaque le standard de bus avionique ARINC 429. Grandement utilisé dans les avions commerciaux, il n'a pas reçu de mise à jour de cybersécurité depuis plus de 30 ans.

On propose en guise de contribution, *onto-by-429*, une ontologie modélisant le standard de bus ARINC 429, afin d'étudier la viabilité de l'explication d'anomalies pour les parties prenantes de l'aviation (pilotes, ouvrier de maintenance, développeurs d'IDS, etc.). Elle a été intégrée à ATMONTO, une ontologie de gestion du trafic aérien produit par la NASA et à l'ontologie MITRE D3FEND qui permet de prendre en compte les concepts de cybersécurité.

Un Engin d'Explication d'Anomalies (EEA) utilise une ontologie interdisciplinaire, liant *onto-by-429*, ATMONTO et D3FEND, afin d'expliquer les anomalies tout en demeurant standardisable, extensible et interopérable. L'ontologie interdisciplinaire prend le nom d'*onto-by-wire*. Elle est implémentée au moyen de la méthodologie SAMOD et assure la création de scénarios de l'attaquant auprès du cadre MITRE ATT&CK Entreprise. Elle est validée lors du processus itératif par un ingénieur spécialisé dans le web sémantique ainsi que par un véritable pilote. *Onto-by-wire* offre donc une représentation de la connaissance qui intègre la gestion du trafic aérien (ATMONTO) et la cyberdéfense (MITRE D3FEND). Pour lier ces deux sous-domaines, on contribue à développer *onto-by-429*, une nouvelle ontologie du standard ARINC 429.

Les scénarios sont basés sur des standards et des protocoles indépendants du manufacturier ou d'un fournisseur. Ce *modus operandi* nous évite de divulguer des secrets d'affaires ou de sécurité tout en nous assurant que l'ontologie soit compatible et extensible selon les besoins de particuliers. Les scénarios ont été également inspirés par les travaux de la littérature. D'une part, le premier scénario est une attaque par usurpation (*spoofing*) de paquets produits par l'ajout d'un LRU malicieux d'une agence gouvernementale. D'autre part, le second scénario est une attaque d'inondation (*flooding*) par déni de service causé par une vulnérabilité perpétrée sur la chaîne d'approvisionnement d'un service d'intelligence. Trois cas d'études prennent ensuite la forme de questions de compétences, à travers des requêtes SPARQL, pour offrir

une explication de la situation avec des recommandations de contre-mesures. Pour s'assurer que les inférences d'*onto-by-wire* soient cohérentes, des tests d'inconsistances aux étapes itératives du développement ont été produits.

7.2 Limites de la solution proposée

Ce travail a révélé la possibilité de produire une ontologie liant la cybersécurité et l'aviation. Néanmoins, la réalisation de ce projet a apporté son lot de défis. Des limites dans l'implémentation de notre approche dans ces domaines sont présentes.

Tout d'abord, il a été retenu de devoir réduire le plus possible le biais de l'expérimentateur. On a toutefois eu des soucis dûs aux accords de non-divulgation avec les partenaires et aux retards d'achats du banc de test causé par la COVID 19. Ces événements ont obligé l'équipe à trouver une solution de remplacement. Des instances de vol produites par la NASA avec ATMONTO, ont pu être acquises [13]. On a évité ainsi de produire nous-mêmes nos données de vol. Par contre au niveau des paquets ARINC 429, n'ayant pas d'ICD pour les produire, on a été contraint de créer des données factices obéissant au standard [27]. Fort heureusement, nous avons pu recevoir vers la fin de nos expériences un ICD de véritables paquets ARINC 429 convertis en paquet AFDX pour une représentation plus réaliste des SDI avec de véritables labels. Les informations de vol n'étant toutefois pas aussi complètes que celles obtenues en liant nos paquets ARINC 429 factices avec ATMONTO, le troisième cas d'utilisation se révèle avoir une intégration moins riche au niveau des facteurs humains que pour les deux premiers.

Ensuite, parce que nos cas d'utilisation ont été inspirés par la littérature, un doute persiste quant à la crédibilité des attaques ainsi que celle des experts. En effet, lors des recherches en cybercriminalité offensive, certaines attaques ne peuvent être complètement divulguées. Certaines font l'objet de censures. De surcroît, même si la crédibilité de l'expert est reconnue en cybersécurité ou en avionique, cela ne signifie pas que ce dernier maîtrise ces deux domaines conjointement [62]. Ainsi, on recommande que davantage de recherche soit conduite pour déterminer le risque de nos vecteurs d'attaques.

Pour finir, le cadre Entreprise de Mitre ATT&CK a été utilisée pour réaliser nos scénarios d'attaque. Les recommandations de notre engin se trouvent être plus génériques en termes de cybersécurité que spécifiquement par rapport à la sécurité des avions. Une approche plus personnalisée intégrant la sécurité de l'avionique est à considérer pour enrichir nos scénarios d'attaque. L'ontologie étant intrinsèquement extensible, il serait possible d'ajouter de nouvelles techniques offensives et défensives avec de nouveaux identifiants sur D3FEND.

7.3 Améliorations futures

On espère que cette preuve de concept pourra offrir diverses améliorations dans des recherches ultérieures. Dans un premier cadre, maintenant qu'elle a offert des résultats autour de l'ARINC 429, nous avons la porte ouverte pour implémenter une ontologie *onto-by-wire* plus sophistiqué. D'autres standards peuvent désormais être amendés afin de lier plus d'un bus avionique. L'ontologie se voulant extensible, *onto-by-AFDX*, *onto-by-1553* et *onto-by-CAN* sont notamment des ajouts qui enrichiraient ce domaine. L'acquisition de partenaire motivé sera un facteur crucial dans la généralisation d'autres attaques de plus grande envergure. On argumente que l'acquisition d'information à propos de la syntaxe des domaines est primordiale et nécessite les fonds et le matériel disponible de prestataires, fabricants et manufacturiers afin de continuer cette recherche.

Dans un second cadre, la méthode d'attribution des poids en contre-défense basée sur la collection de techniques d'attaque du MITRE est choisie en fonction d'un processus de décision markovien. Les contre-mesures les plus recommandées ne prendront donc pas forcément les techniques dans leur ensemble. On juge pertinent dans un cadre futur d'intégrer davantage de techniques de traitement automatique des langues (NLP) et d'extraction d'information pour ainsi optimiser l'explication des techniques désignées entre l'humain et l'avion et pour formuler convenablement de meilleurs modèles de prise de décisions.

Dans un troisième cadre, notre travail a permis un pont entre l'explication des résultats vers un humain. Toutefois, il serait pertinent d'approfondir ce pont à travers les recherches du Explainable AI (XAI) sur la relation d'explication. La prédictibilité de l'IA demeure encore un sujet de recherche à long terme en soi. Aborder les facteurs humains dans les mesures de mitigation pour préparer un équipage de vol à faire face à un événement offensif est pertinent dans le développement des mécanismes de défense et des mesures d'urgence. La Defense Advanced Research Projects Agency (DARPA), une agence dans le département américain de la Défense, tente de répondre justement ce besoin de développer des mécanismes d'explication pour le domaine critique de la sécurité [72]. Comme montré au niveau de la médecine, l'ontologie peut aider dans l'explication de diagnostic. On y chercher à éviter des réponses non traçables d'un processus de boîte noire, tel que vu que chez les méthodes d'apprentissage machine [73]. Combiner l'ontologie avec le XAI permettrait ainsi une meilleure consistance des résultats de modèle de probabilité en IA tout en combinant une syntaxe chez les modèles entrainés.

Finalement, notre laboratoire vient d'acquérir un simulateur de vol dans le but d'enrichir ses expériences. La portée de notre recherche serait dans un avenir proche capable de produire

nos propres cas d'utilisation à travers ce simulateur. Il permettrait aussi de combiner les données produites en gestion du trafic aérien avec celles des bus avioniques. Notre volonté serait d'y lier l'explorateur d'anomalie. Outre cela, on envisage la possibilité d'intégrer un IDS et ensuite interpréter de véritables attaques détectées.

À travers l'ambition à laquelle notre laboratoire aspire en cybersécurité de l'avionique, nous invitons tous les chercheurs intéressés par ce domaine ou à consulter notre ontologie. Votre collaboration ainsi que vos idées sont la bienvenue à gigl-ES@polymtl.ca.

RÉFÉRENCES

[1] A.-K. International Civil Aviation Organization (ICAO), “Convention on Civil Aviation ("Chicago Convention"),” *7 December 1944*, p. 28, avr. 1947. [En ligne]. Disponible : https://www.refworld.org/docid/3ddca0dd4.html

[2] J. de Ferrari et E. Laughlin, “GAO-21-86, FAA should fully implement key practices to strenghten its oversight of avionics risks,” United States Government Accountability Office, US, Aviation Cybersecurity GAO-21-86, oct. 2020.

[3] A. Uncu, S. Üzümcü et A. A. Mert, “Cyber security concerns regarding federated, partly ima and full ima implementations,” dans *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, 2019, p. 1–5.

[4] M. Monteiro *et al.*, “An integrated mission and cyber simulation for air traffic control,” nov. 2016. [En ligne]. Disponible : https://doi.org/10.1109/itsc.2016.7795988

[5] E. Papadimitriou, “Transport safety and human factors in the era of automation_ What can transport modes learn from each other ?” *Accident Analysis and Prevention*, p. 16, 2020.

[6] N. G. Markevich et A. Wool, “Hardware Fingerprinting for the ARINC 429 Avionic Bus [W],” *arXiv :2003.12456 [cs]*, mars 2020, arXiv : 2003.12456. [En ligne]. Disponible : http://arxiv.org/abs/2003.12456

[7] M. Strohmeier *et al.*, “Building an avionics laboratory for cybersecurity testing,” dans *Proceedings of the 15th Workshop on Cyber Security Experimentation and Test*, ser. CSET ’22. New York, NY, USA : Association for Computing Machinery, 2022, p. 10–18. [En ligne]. Disponible : https://doi.org/10.1145/3546096.3546101

[8] L. Ryon et G. Rice, “A Safety-focused Security Risk Assessment of Commercial Aircraft Avionics [w],” dans *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*. London : IEEE, sept. 2018, p. 1–8. [En ligne]. Disponible : https://ieeexplore.ieee.org/document/8569314/

[9] L. Ryon, G. Rice et J. Potts, “An Avionics Cyber Intrusion Detection System,” dans *AIAA Scitech 2020 Forum*. Orlando, FL : American Institute of Aeronautics and Astronautics, janv. 2020. [En ligne]. Disponible : https://arc.aiaa.org/doi/10.2514/6.2020-0318

[10] J.-Y. D. Miceli, “Détection d’attaques informatiques sophistiquées contre les communications ADS-B en aviation [W],” p. 108, 2020.

[11] A. Canito *et al.*, “An ontology to promote interoperability between cyber-physical security systems in critical infrastructures,” dans *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, 2020, p. 553–560.

[12] C. C. Insaurralde et E. Blasch, “Situation Awareness Decision Support System for Air Traffic Management Using Ontological Reasoning,” *Journal of Aerospace Information Systems*, vol. 19, n°. 3, p. 224–245, mars 2022. [En ligne]. Disponible : https://arc.aiaa.org/doi/10.2514/1.I010989

[13] R. M. Keller, “ATMONTO Ontology,” mars 2018. [En ligne]. Disponible : https://data.nasa.gov/ontologies/atmonto/

[14] P. E. Kaloroumakis et M. J. Smith, “Toward a Knowledge Graph of Cybersecurity Countermeasures,” p. 11, 2021.

[15] E. Levy *et al.*, “Anomili : Spoofing prevention and explainable anomaly detection for the 1553 military avionic bus,” 2022. [En ligne]. Disponible : https://arxiv.org/abs/2202.06870

[16] L. Obrst, P. Chase et R. Markeloff, “Developing an ontology of the cyber security domain,” dans *STIDS*, 2012.

[17] M. Bozdal *et al.*, “Evaluation of can bus security challenges,” *Sensors*, vol. 20, p. 16–17, 04 2020.

[18] S. Peroni, “A simplified agile methodology for ontology development,” dans *OWL : Experiences and Directions – Reasoner Evaluation*, M. Dragoni, M. Poveda-Villalón et E. Jimenez-Ruiz, édit. Cham : Springer International Publishing, 2017, p. 55–69.

[19] L. Bogoda, J. Mo et C. Bil, “A Systems Engineering Approach To Appraise Cybersecurity Risks Of CNS/ATM and Avionics Systems,” dans *2019 Integrated Communications, Navigation and Surveillance Conference (ICNS)*. Herndon, VA, USA : IEEE, avr. 2019, p. 1–15. [En ligne]. Disponible : https://ieeexplore.ieee.org/document/8735376/

[20] E. Blasch, “Ontologies for nextgen avionics systems,” dans *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*. IEEE, sept. 2015. [En ligne]. Disponible : https://doi.org/10.1109/dasc.2015.7311395

[21] E. Blasch, R. Sabatini et A. Roy, “Cyber Awareness Trends in Avionics [W],” p. 8, 2021.

[22] J. Graham *et al.*, “Design of a single pilot cockpit for airline operations,” dans *2014 Systems and Information Engineering Design Symposium (SIEDS)*, 2014, p. 210–215.

[23] M. Wolf, M. Minzlaff et M. Moser, “Information Technology Security Threats to Modern e-Enabled Aircraft : A Cautionary Note,” *Journal of Aerospace*

Information Systems, vol. 11, n°. 7, p. 447–457, juill. 2014. [En ligne]. Disponible : https://arc.aiaa.org/doi/10.2514/1.I010156

[24] A. Lomas, "Def con 28 : Introduction to acars," 2020. [En ligne]. Disponible : https://www.pentestpartners.com/security-blog/introduction-to-acars/

[25] P. Skaves, "Faa aircraft systems information security protection (asisp) overview, paper 132," dans *2015 Integrated Communication, Navigation and Surveillance Conference (ICNS)*, 2015, p. 1–41.

[26] A. LAMBREGTS, *Vertical flight path and speed control autopilot design using total energy principles*. American Institute of Aeronautics and Astronautics, août 1983. [En ligne]. Disponible : https://doi.org/10.2514/6.1983-2239

[27] P. Frodyma et B. Waldman, "ARINC 429 Specification Tutorial," 2010, 2.1 edition.

[28] P. Kiley, "Investigating CAN Bus Network Integrity in Avionics System," 2019. [En ligne]. Disponible : https://www.rapid7.com/research/report/investigating-can-bus-network-integrity-in-avionics-systems/

[29] Liansheng Liu et Lei Zhang, "Research of method for testing ARINC429 bus [W]," dans *2010 Chinese Control and Decision Conference*. Xuzhou, China : IEEE, mai 2010, p. 2203–2208. [En ligne]. Disponible : http://ieeexplore.ieee.org/document/5498854/

[30] D. D. Santo *et al.*, "Exploiting the mil-std-1553 avionic data bus with an active cyber device," *Comput. Secur.*, vol. 100, p. 102097, 2021.

[31] C. Fuchs, "The evolution of avionics networks from arinc 429 to afdx," *Innovative Internet Technologies and Mobile Communications and Aerospace Networks (IITM and AN)*, vol. 65, 01 2012.

[32] F. Van Harmelen, V. Lifschitz et B. Porter, *Handbook of knowledge representation*. Elsevier, 2008.

[33] G. Antoniou et F. Van Harmelen, *A semantic web primer*. MIT press, 2004.

[34] O. Stan *et al.*, "Protecting military avionics platforms from attacks on mil-std-1553 communication bus," 2017. [En ligne]. Disponible : https://arxiv.org/abs/1707.05032

[35] E. Gringinger, D. Eier et D. Merkl, "Ontology-based CNS software development," mai 2010. [En ligne]. Disponible : https://doi.org/10.1109/icnsurv.2010.5503306

[36] R. M. Keller, "Ontologies for aviation data management," dans *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, 2016, p. 1–9.

[37] S. Koo et S. W. Kwong, "Teaching computer communication networks : Top-down or bottom-up ?" dans *Proceedings Frontiers in Education 35th Annual Conference*. IEEE. [En ligne]. Disponible : https://doi.org/10.1109/fie.2005.1612250

[38] F. Alrefaei *et al.*, “Cyber physical systems, a new challenge and security issue for the aviation,” dans *2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, 2021, p. 1–5.

[39] C. Xenofontos *et al.*, “Consumer, commercial and industrial iot (in)security : Attack taxonomy and case studies,” 05 2021.

[40] A. Babar *et al.*, “An approach to represent and transform application-specific constraints for an intrusion detection system,” dans *Proceedings of the 30th Annual International Conference on Computer Science and Software Engineering*, ser. CASCON ’20. USA : IBM Corp., 2020, p. 53–62.

[41] H. Sedjelmaci et S.-M. Senouci, “Cyber security methods for aerial vehicle networks : taxonomy, challenges and solution,” *The Journal of Supercomputing*, vol. 74, p. 4928–4944, 2018.

[42] O. Stan *et al.*, “Intrusion detection system for the mil-std-1553 communication bus,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, n°. 4, p. 3010–3027, 2020.

[43] R. Yahalom *et al.*, “Datasets of rt spoofing attacks on mil-std-1553 communication traffic,” *Data in Brief*, vol. 23, p. 103863, 2019. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S2352340919302148

[44] J. Ryan, “Cyber security : The mess we’re in and why it’s going to get worse,” *CSPRI Report*, p. 1–13, 2011.

[45] H. Teso, “Aircraft hacking - practical aero series,” 04 2013.

[46] S. A. Kumar et B. Xu, “Vulnerability Assessment for Security in Aviation Cyber-Physical Systems [W],” dans *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*. New York, NY, USA : IEEE, juin 2017, p. 145–150. [En ligne]. Disponible : http://ieeexplore.ieee.org/document/7987190/

[47] S. Crow *et al.*, “Triton : A Software-Reconfigurable federated avionics testbed,” dans *12th USENIX Workshop on Cyber Security Experimentation and Test (CSET 19)*. Santa Clara, CA : USENIX Association, août 2019. [En ligne]. Disponible : https://www.usenix.org/conference/cset19/presentation/crow

[48] W. True *et al.*, “Cybersecurity for Flight Deck Data Exchange,” dans *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*. San Antonio, TX, USA : IEEE, oct. 2021, p. 1–13. [En ligne]. Disponible : https://ieeexplore.ieee.org/document/9594376/

[49] A. Lomas, “Def con 30. hacking efbs. engine performance,” 2022. [En ligne]. Disponible : https://www.pentestpartners.com/security-blog/def-con-30-hacking-efbs-engine-performance/

[50] S. Shappell et D. Wiegmann, “The human factors analysis and classification system-hfacs,” 01 2000.

[51] C. Eidsness, “Cryptographic Signing of ARINC-429 Data,” p. 6, 2022.

[52] F. Pascale *et al.*, “Cybersecurity in Automotive : An Intrusion Detection System in Connected Vehicles,” *Electronics*, vol. 10, nº. 15, p. 1765, juill. 2021. [En ligne]. Disponible : https://www.mdpi.com/2079-9292/10/15/1765

[53] L.-P. Morel, “Using Ontologies to Detect Anomalies in the Sky (Masters thesis École Polytechnique de Montréal)),” p. 90, 2017.

[54] M. Wrana, “Avionics Network Anomaly Detection through True-Skip Learning,” 2022.

[55] J. Undercoffer, A. Joshi et J. Pinkston, “Modeling computer attacks : An ontology for intrusion detection,” dans *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2003, p. 113–135. [En ligne]. Disponible : https://doi.org/10.1007/978-3-540-45248-5_7

[56] R. Y. Venkata, P. Kamongi et K. Kavi, “An ontology-driven framework for security and resiliency in cyber physical systems,” vol. 2018, 2018, p. 23.

[57] A. M. Shaaban *et al.*, “Ontology-Based Model for Automotive Security Verification and Validation,” dans *Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services*. Munich Germany : ACM, déc. 2019, p. 73–82. [En ligne]. Disponible : https://dl.acm.org/doi/10.1145/3366030.3366070

[58] D. Stefanidis *et al.*, “The ICARUS ontology,” juin 2020. [En ligne]. Disponible : https://doi.org/10.1145/3405962.3405983

[59] C. C. Insaurralde *et al.*, “Uncertainty Considerations for Ontological Decision-Making Support in Avionics Analytics [W],” dans *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*. London : IEEE, sept. 2018, p. 1–9. [En ligne]. Disponible : https://ieeexplore.ieee.org/document/8569816/

[60] C. C. Insaurralde et E. Blasch, “Trust Evaluation of Ontological Decision Support Systems for Avionics Analytics [W],” dans *2021 Integrated Communications Navigation and Surveillance Conference (ICNS)*. Dulles, VA, USA : IEEE, avr. 2021, p. 1–10. [En ligne]. Disponible : https://ieeexplore.ieee.org/document/9441570/

[61] L. Palacios *et al.*, “Avionics Maintenance Ontology Building for Failure Diagnosis Support :,” dans *Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*. Porto, Portugal : SCITEPRESS - Science and and Technology Publications, 2016, p. 204–209. [En ligne]. Disponible : http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0006092002040209

[62] E. Habler, R. Bitton et A. Shabtai, “Evaluating the security of aircraft systems,” *arXiv preprint arXiv :2209.04028*, 2022.

[63] X. Hu et J. Liu, “Requirements knowledge model construction and requirements elicitation method of avionics systems software based on multi-ontology,” dans *2021 23rd International Conference on Advanced Communication Technology (ICACT)*, 2021, p. 1–15.

[64] P. Skaves, “Information for cyber security issues related to aircraft systems,” dans *2011 IEEE/AIAA 30th Digital Avionics Systems Conference*, 2011, p. 1C2–1–1C2–11.

[65] R. S. Mueller et M. W. A. Cat, *Report on the investigation into Russian interference in the 2016 presidential election*. US Department of Justice Washington, DC, 2019, vol. 1.

[66] K. T. Smith *et al.*, “Cyber terrorism cases and stock market valuation effects,” *Information & Computer Security*, 2023.

[67] W. Xiong *et al.*, “Cyber security threat modeling based on the mitre enterprise att&ck matrix,” *Software and Systems Modeling*, vol. 21, 02 2022.

[68] M. Debellis, “A practical guide to building owl ontologies using protégé 5.5 and plugins,” *The university of Manchester*, 04 2021.

[69] J. Goh et D. Wiegmann, “Human factors analysis of accidents involving visual flight rules flight into adverse weather,” *Aviation, space, and environmental medicine*, vol. 73, p. 817–22, 09 2002.

[70] L. Irshad *et al.*, “Using Rio-Paris Flight 447 Crash to Assess Human Error and Failure Propagation Analysis Early in Design,” *ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg*, vol. 6, n°. 1, 11 2019, 011008. [En ligne]. Disponible : https://doi.org/10.1115/1.4044790

[71] P. Berthier, J. M. Fernandez et J.-M. Robert, “Sat : Security in the air using tesla,” *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*, p. 1–10, 2017.

[72] J. Gilbert, “Explainable ids for dos attacks,” Mémoire de maîtrise, Polytechnique Montréal, December 2020. [En ligne]. Disponible : https://publications.polymtl.ca/5582/

[73] C. Panigutti, A. Perotti et D. Pedreschi, “Doctor xai : an ontology-based approach to black-box sequential data classification explanations,” dans *Proceedings of the 2020 conference on fairness, accountability, and transparency*, 2020, p. 629–639.

ANNEXE A ANNEXE A

```
1
2 CONSTRUCT
3 {?DefenseTechniques rdfs:numberOfCounterMeasures ?totalDefenseTechniques.}
4
5 WHERE{{
6
7 SELECT ?DefenseTechniques  (COUNT(?DefenseTechniques) as ?totalDefenseTechniques) WHERE {
8
9     #Get all ATTACK Techniques from a Scenario A
10    <https://data.nasa.gov/ontologies/atmontoCore#ScenarioAttack1> a d3fend:ScenarioATTACK;
11            d3fend:first-ATTACK-scenario ?FirstAttackTechnique.
12    ?FirstAttackTechnique d3fend:next-ATTACK-scenario* ?AttackTechniques.
13    ?AttackTechniques d3fend:attack-id ?AttackID.
14
15     #Matches Attacks Techniques from the scenario with MITRE ATT&CK ones
16    ?OffensiveTechniques d3fend:attack-id ?AttackID.
17
18    #Lists All Defense Technique from MITRE D3FEND
19  ?DefenseTechniques d3fend:d3fend-id ?DefenseID.
20
21     #Gets all Defense Techniques propise with the attacked artefact
22    ?OffensiveTechniques d3fend:d3fend-object-property ?Artefact.
23    ?Artefact rdfs:subClassOf* ?Artefacts.
24    ?DefenseTechniques d3fend:d3fend-object-property ?Artefacts.
25
26
27    }GROUP BY ?DefenseTechniques
28   HAVING (?totalDefenseTechniques > 3)
29        ORDER BY DESC (?totalDefenseTechniques)
30
31 }}
```

Listing A.1 Requête SPARQL de construction du scénario d'attaque A

ANNEXE B ANNEXE B

```
1
2  SELECT DISTINCT
3  ?Warning ?Timestamp ?AttackScenario ?AttackVector
4  ?CompromisedDevice ?ErroneousParameter ?DataReceived
5  ?DataExpected ?CounterMeasure ?CounterMeasureDetail
6
7  WHERE {
8
9      ### WARNING ###
10     BIND (IF(?countAltitudeValue > 1, "Warning: More than one altitude value", "OK") as ?Warning)
11     #Display warning based on the number of value
12
13     {
14         SELECT DISTINCT
15         (count(*) as ?countAltitudeValue) ?Timestamp
16         ?AttackScenario ?AttackVector ?CompromisedDevice
17         ?ErroneousParameter ?DataReceived  ?DataExpected
18         ?CounterMeasure ?CounterMeasureDetail
19
20             WHERE{
21
22         ##############################
23         ###### MITRE ATT&CK #########
24         ##############################
25
26         #Get all ATTACK Techniques from a Scenario
27         ?ScenarioAtk1 a d3fend:ScenarioATTACK;
28                       d3fend:scenarioDescription ?AttackScenario;
29                 d3fend:first-ATTACK-scenario ?FirstAttackTechnique.
30
31         ?FirstAttackTechnique d3fend:next-ATTACK-scenario*/d3fend:attack-id ?AttackID.
32
33         #Matches Attacks Techniques from the scenario with
34         #MITRE ATT&CK ones
35         ?OffensiveTechniques d3fend:attack-id ?AttackID;
36                              rdfs:label ?Attack.
37
38         ##############################
39         ###### MITRE D3FEND ########
40         ##############################
41
42         #Filter Defense Technique with the counter
43         #number of scenario A
44         ?DefenseTechniques rdfs:numberOfCounterMeasures ?PriorityValue.
45         FILTER (?PriorityValue > 1)
46
47         #Lists All Defense Technique from MITRE D3FEND
48         ?DefenseTechniques d3fend:d3fend-id ?DefenseID;
49           rdfs:label ?CounterMeasure;
50             d3fend:definition ?CounterMeasureDetail.
51
52         #Gets all Defense Techniques propise with the attacked
53         #artefact
54         ?OffensiveTechniques d3fend:d3fend-object-property/rdfs:subClassOf* ?Artefacts.
55         ?DefenseTechniques d3fend:d3fend-object-property ?Artefacts.
56         ?Artefacts rdfs:label ?ArtefactAffected.
57
58         ######################################
59         ##### Airport Traffic Management ####
60         ######################################
61         ?Flight a atm:Flight;
62                   atm:hasActualRoute ?ActualRoute;
63                   atm:departureAirport ?Airport;
64                   atm:aircraftFlown ?Aircraft.
65
66         #Transponder uses ADS-B compatible with mode S
67         ?Aircraft eqp:modeSCode ?ModeSCode.
68         BIND(STRLEN(?ModeSCode) AS ?len_o)
```

```
69          #Verify if the string is not NULL or empty
70
71          #Obtains all Trackpoints from a Route
72          ?ActualRoute gen:hasFirstItem/gen:hasNextItem* ?NextTrackpoint.
73
74          ?NextTrackpoint fbw:hasFlightPhase ?FlightPhase;
75                          #For ARINC429 Data
76                          fbw:hasCommunicationFix/fbw:sendsFlyByWireMessage ?FlyByWireFix;
77                          #Altitude in a Trackpoint
78                          atm:aircraftFix/gen:altitude ?DataExpected;
79                          rdfs:label ?Timestamp.
80
81          #Flight Phase scenario: Landing.
82          ?FlightPhase fbw:flightPhaseAcronym ?FlightPhaseAcronym;
83                       fbw:flightPhaseTerm ?FlightPhaseTerm.
84          FILTER (?FlightPhaseTerm =  "Landing" || ?FlightPhaseAcronym = "LDG")
85
86          ##############################
87          ###### ARINC 429 Messages ##
88          ##############################
89          # Attack Vector
90          BIND (exists{?FlyByWireFix rdf:type fbw:ARINC429Message}AS ?isARINC429)
91          #Display warning based on the number of value
92          BIND (IF(?isARINC429, "Compromised ARINC 429 Bus", "Compromised Bus") as ?AttackVector)
93
94          #ARINC429 - Label : 370 Source Altmeter
95          ?FlyByWireFix fbw:hasARINC429Datafield ?Label.
96
97          ?Label fbw:LSB ?LSBLabel;
98                 fbw:MSB ?MSBLabel;
99                 fbw:datafieldValue ?LabelNumber;
100                fbw:decodedInformation ?ErroneousParameter.
101         FILTER((?LSBLabel = 1 || ?MSBLabel = 8) && ?LabelNumber = 370)
102
103         #ARINC429 - Data : Altimeter Value
104         ?FlyByWireFix fbw:hasARINC429Datafield ?DataValue.
105         ?DataValue fbw:datafieldValue ?DataReceived;
106                    fbw:LSB ?DataLSB;
107                    fbw:MSB ?DataMSB;
108                    fbw:decodedInformation ?DataCaracteristic.
109         FILTER (?DataLSB = 11 || ?DataMSB = 19)
110
111         # ARINC 429 - SDI Source: Radio Altimeter
112         ?FlyByWireFix fbw:hasARINC429Datafield ?SDI.
113         ?SDI fbw:LSB ?SDILSB;
114              fbw:MSB ?SDIMSB;
115              fbw:hasSDISourceORDestination/fbw:isLRUOf/rdfs:label ?CompromisedDevice.
116              FILTER (?SDILSB = 9 || ?SDIMSB = 10)
117
118         ##################################
119         ##### WEATHER CONDITION #######
120         ##################################
121         ?Airport data:hasMETARreport ?METARreport.
122         ?METARreport data:hasWeatherCondition ?METARWeather;
123                      data:hasVisibilityCondition ?METARVisibility.
124
125         #Heavy weather increase pilot's workload
126         ?METARWeater data:weatherIntensity ?WeatherIntensity
127         FILTER (?WeatherIntensity = "heavy")
128
129         #Lack of visibility increase pilot's workload
130         ?METARVisibility data:unlimitedVisibility ?IsVisible
131         FILTER (?IsVisible = false)
132
133     }
134     GROUP BY  ?Timestamp ?AttackScenario ?AttackVector
135     ?CompromisedDevice ?ErroneousParameter ?DataReceived
136     ?DataExpected ?CounterMeasure ?CounterMeasureDetail
137     }
138 }
```

Listing B.1 Requête SPARQL du premier cas d'étude

```
1
2 SELECT DISTINCT ?Warning ?Timestamp ?AttackScenario ?AttackVector ?CompromisedDevice ?ErroneousParameter
      ?DataReceived ?DataExpected ?CounterMeasure ?CounterMeasureDetail
3  WHERE{
4
5      ##################
6        #### Warning ####
7        ##################
8
9        BIND ( "Warning: Anomalous Ground Speed" as ?Warning) #Display warning
10
11       ##############################
12     ######## MITRE ATT&CK #########
13     ##############################
14
15       #Get all ATTACK Techniques from a Scenario
16       <https://data.nasa.gov/ontologies/atmontoCore#ScenarioAttack2> a d3fend:ScenarioATTACK;
17                       d3fend:scenarioDescription ?AttackScenario;
18                d3fend:first-ATTACK-scenario ?FirstAttackTechnique.
19
20       ?FirstAttackTechnique d3fend:next-ATTACK-scenario*/d3fend:attack-id ?AttackID.
21
22       #Matches Attacks Techniques from the scenario with MITRE ATT&CK ones
23       ?OffensiveTechniques d3fend:attack-id ?AttackID;
24                            rdfs:label ?Attack.
25
26       ##############################
27       ###### MITRE D3FEND ########
28       ##############################
29
30      #Filter Defense Technique with the counter #number of scenario B
31       ?DefenseTechniques rdfs:numberOfCounterMeasures ?PriorityValue;
32                          rdfs:subClassOf/d3fend:accesses ?Model.
33       ?Model a d3fend:DefensiveTactic;
34                rdfs:label ?DefenseTacticLabel.
35
36       FILTER (?PriorityValue > 4)
37
38       #Lists All Defense Technique from MITRE D3FEND
39       ?DefenseTechniques d3fend:d3fend-id ?DefenseID;
40           rdfs:label ?CounterMeasureLabel;
41           d3fend:definition ?CounterMeasureDetail.
42
43           BIND (CONCAT (
44           ?DefenseTacticLabel, ": ",?DefenseID, " - ", ?CounterMeasureLabel
45         )as ?CounterMeasure)
46
47       #Gets all Defense Techniques propise with the attacked artefact
48       ?OffensiveTechniques d3fend:d3fend-object-property/rdfs:subClassOf* ?Artefacts.
49       ?DefenseTechniques d3fend:d3fend-object-property ?Artefacts.
50       ?Artefacts rdfs:label ?ArtefactAffected.
51
52    ######################################
53    ##### Airport Traffic Management ####
54    ######################################
55    ?Flight a atm:Flight;
56              atm:hasActualRoute ?ActualRoute;
57              atm:departureAirport ?Airport.
58
59      #Obtains all Trackpoints from a Route
60        ?ActualRoute gen:hasFirstItem/gen:hasNextItem* ?Trackpoints.
61
62    ?TrackPoints fbw:hasFlightPhase ?FlightPhase;
63                 fbw:hasCommunicationFix ?CommunicationFix; #For ARINC 429 Messages
64                 atm:groundSpeed ?DataExpected; #Ground speed at the TrackPoint
65               rdfs:label ?Timestamp.
66
67      #Flight Phase scenario: Take-Off.
68    ?FlightPhase fbw:flightPhaseAcronym ?FlightPhaseAcronym;
69                 fbw:flightPhaseTerm ?FlightPhaseTerm.
70    FILTER (?FlightPhaseTerm =  "Takeoff" ||
71          ?FlightPhaseAcronym = "TOF")
72
73    ##############################
```

```
74      ###### ARINC 429 Messages ###
75      ##############################
76
77      ?CommunicationFix fbw:sendsFlyByWireMessage ?FlyByWireFix.
78      ?FlyByWireFix fbw:hasARINC429Datafield ?Label.
79
80          # Attack Vector
81          BIND (exists{?FlyByWireFix rdf:type fbw:ARINC429Message} AS ?isARINC429)
82
83          #Display warning based on the number of value
84          BIND (IF(?isARINC429, "Compromised ARINC 429 Bus", "Compromised Bus") as ?AttackVector)
85
86      #ARINC429 - Label : 174 GroundSpeed
87      ?Label fbw:LSB ?LSBLabel;
88             fbw:MSB ?MSBLabel;
89             fbw:datafieldValue ?LabelNumber;
90               fbw:decodedInformation ?ErroneousParameter.
91      FILTER((?LSBLabel = 1 || ?MSBLabel = 8) && ?LabelNumber = 174)
92
93      #ARINC429 - Data : Speed Value
94      ?FlyByWireFix fbw:hasARINC429Datafield ?DataValue.
95      ?DataValue fbw:datafieldValue ?DataReceived;
96                           fbw:LSB ?DataLSB;
97                           fbw:MSB ?DataMSB;
98                           fbw:decodedInformation ?DataCaracteristic
99                     FILTER (?DataLSB = 11 || ?DataMSB = 19)
100
101         # ARINC 429 - SDI Source: Radio Altimeter
102         ?FlyByWireFix fbw:hasARINC429Datafield ?SDI.
103         ?SDI fbw:LSB ?SDILSB;
104              fbw:MSB ?SDIMSB;
105              fbw:hasSDISourceORDestination/fbw:isLRUOf/rdfs:label ?CompromisedDevice.
106         FILTER (?SDILSB = 9 || ?SDIMSB = 10)
107
108
109     ################################
110     ##### WEATHER CONDITION #######
111     ################################
112     ?Airport data:hasMETARreport ?METARreport.
113     ?METARreport data:hasWeatherCondition ?METARWeather;
114              data:hasVisibilityCondition ?METARVisibility.
115
116     #Heavy weather increase pilot's workload
117     ?METARWeater data:weatherIntensity ?WeatherIntensity
118     FILTER (?WeatherIntensity = "heavy")
119
120     #Lack of visibility increase pilot's workload
121     ?METARVisibility data:unlimitedVisibility ?IsVisible
122     FILTER (?IsVisible = false)
123
124
125     FILTER (ABS(?DataExpected - ?DataReceived) > 20)
126   }
127 }
```

Listing B.2 Requête SPARQL du deuxième cas d'étude

```
1
2 PREFIX d3fend: <http://d3fend.mitre.org/ontologies/d3fend.owl#>
3 SELECT DISTINCT ?Warning ?Timestamp ?AttackScenario ?AttackVector ?CompromisedDevice ?ErroneousParameter
      ?DataReceived ?SSMaracteristic ?CounterMeasure ?CounterMeasureDetail
4   WHERE{
5
6       ##################
7         #### Warning ####
8         ##################
9         BIND ( "Warning: Anomalous Auto Pilot Disabled" as ?Warning) #Display warning
10
11        ##############################
12      ######## MITRE ATT&CK #########
13      ##############################
14
15        #Get all ATTACK Techniques from a Scenario
16        <https://data.nasa.gov/ontologies/atmontoCore#ScenarioAttack1> a d3fend:ScenarioATTACK;
```

```
                        d3fend:scenarioDescription ?AttackScenario;
                d3fend:first-ATTACK-scenario ?FirstAttackTechnique.

        ?FirstAttackTechnique d3fend:next-ATTACK-scenario*/d3fend:attack-id ?AttackID.

        #Matches Attacks Techniques from the scenario with MITRE ATT&CK ones
        ?OffensiveTechniques d3fend:attack-id ?AttackID;
                             rdfs:label ?Attack.

        ##############################
        ###### MITRE D3FEND ########
        ##############################

       #Filter Defense Technique with the counter #number of scenario A
        ?DefenseTechniques rdfs:numberOfCounterMeasures ?PriorityValue;
                           rdfs:subClassOf/d3fend:accesses ?Model.
        ?Model a d3fend:DefensiveTactic;
                rdfs:label ?DefenseTacticLabel.

        FILTER (?PriorityValue > 3)
      FILTER ( ?DefenseTacticLabel = "Detect")

        #Lists All Defense Technique from MITRE D3FEND
        ?DefenseTechniques d3fend:d3fend-id ?DefenseID;
            rdfs:label ?CounterMeasureLabel;
            d3fend:definition ?CounterMeasureDetail.

       BIND (CONCAT (?DefenseTacticLabel, ": ",?DefenseID, " - ", ?CounterMeasureLabel
       ) as ?CounterMeasure)

        #Gets all Defense Techniques propise with the attacked artefact
        ?OffensiveTechniques d3fend:d3fend-object-property/rdfs:subClassOf* ?Artefacts.
        ?DefenseTechniques d3fend:d3fend-object-property ?Artefacts.
        ?Artefacts rdfs:label ?ArtefactAffected.

    ######################################
    ##### Airport Traffic Management ####
    ######################################
    ?Flight a atm:Flight;
              atm:hasActualRoute ?ActualRoute;
              atm:departureAirport ?Airport.

      #Obtains all Trackpoints from a Route
        ?ActualRoute gen:hasFirstItem/gen:hasNextItem* ?Trackpoints.

    ?TrackPoints fbw:hasFlightPhase ?FlightPhase;
                 fbw:hasCommunicationFix ?CommunicationFix; #For ARINC 429 Messages
             rdfs:label ?Timestamp.

      #Flight Phase scenario: En Route
    ?FlightPhase fbw:flightPhaseAcronym ?FlightPhaseAcronym;
                 fbw:flightPhaseTerm ?FlightPhaseTerm.
    FILTER (?FlightPhaseTerm =  "En Route" || ?FlightPhaseAcronym = "ENR")

      #Auto-Pilot exists and is Disabled
      ?AFCS a fbw:AFCS;
              fbw:hasAutoPilotEnabled ?APMode.
      FILTER (?APMode = false)

    ############################
    ###### ARINC 429 Messages ##
    ############################
    ?CommunicationFix fbw:sendsFlyByWireMessage ?FlyByWireFix.
    ?FlyByWireFix fbw:hasARINC429Datafield ?Label.

        # Attack Vector
        BIND (exists{?FlyByWireFix rdf:type fbw:ARINC429Message} AS ?isARINC429)
        #Display warning based on the number of value
        BIND (IF(?isARINC429, "Compromised ARINC 429 Bus", "Compromised Bus") as ?AttackVector)

    #ARINC429 - Label : 008 Autopilot Status
    ?Label fbw:LSB ?LSBLabel;
           fbw:MSB ?MSBLabel;
           fbw:datafieldValue ?LabelNumber;
```

```
91              fbw:decodedInformation ?ErroneousParameter.
92      FILTER((?LSBLabel = 1 || ?MSBLabel = 8) && ?LabelNumber = 8)
93
94      #ARINC429 - Data : Autopilot Data
95      ?FlyByWireFix fbw:hasARINC429Datafield ?DataValue.
96      ?DataValue fbw:decodedInformation ?DataReceived;
97                          fbw:LSB ?DataLSB;
98                          fbw:MSB ?DataMSB;
99                          fbw:decodedInformation ?DataCaracteristic
100       FILTER (?DataLSB = 11 || ?DataMSB = 19)
101
102       #ARINC429 - SSM : Critical in Normal Operation
103     ?FlyByWireFix fbw:hasARINC429Datafield ?SSMValue.
104     ?SSMValue fbw:decodedInformation ?SSMaracteristic;
105                         fbw:LSB ?SSMLSB;
106                         fbw:MSB ?SSMMSB;
107                         fbw:datafieldValue ?SSMstatus.
108                       FILTER ((?SSMLSB = 30 || ?SSMMSB = 31) && ?SSMstatus = 0)
109
110         # ARINC 429 - SDI Source: Radio Altimeter
111         ?FlyByWireFix fbw:hasARINC429Datafield ?SDI.
112         ?SDI fbw:LSB ?SDILSB;
113               fbw:MSB ?SDIMSB;
114               fbw:hasSDISourceORDestination/fbw:isLRUOf/rdfs:label ?CompromisedDevice.
115         FILTER (?SDILSB = 9 || ?SDIMSB = 10)
116   }
117 }
```

Listing B.3 Requête SPARQL du troisième cas d'étude

ANNEXE C ANNEXE C

```
1
2  Declaration(Class(fbw:ARINC429))
3  Declaration(Class(fbw:ARINC429Message))
4  Declaration(Class(fbw:ARINC429MessageBCD))
5  Declaration(Class(fbw:SSM))
6  Declaration(Class(fbw:SSMFailureWarning))
7  Declaration(Class(fbw:SSMFunctionalTest))
8  Declaration(Class(fbw:SSMNegativeDataValue))
9  Declaration(Class(fbw:SSMPositiveDataValue))
10
11 ##############################
12 #    Data Properties
13 ##############################
14
15 # Data Property: fbw:dataTypeARINC429 (fbw:dataTypeARINC429)
16
17 AnnotationAssertion(rdfs:comment fbw:dataTypeARINC429 "The type of ARINC 429
18 data which is generally : BCD, BCR or Discrete"@en)
19 SubDataPropertyOf(fbw:dataTypeARINC429 fbw:fbw-data-property)
20 FunctionalDataProperty(fbw:dataTypeARINC429)
21 DataPropertyDomain(fbw:dataTypeARINC429 fbw:ARINC429Message)
22 DataPropertyDomain(fbw:dataTypeARINC429 fbw:AvionicsBusMessage)
23 DataPropertyDomain(fbw:dataTypeARINC429 ObjectIntersectionOf(fbw:ARINC429Message DataSomeValuesFrom(fbw:
       dataTypeARINC429 xsd:string)))
24 DataPropertyDomain(fbw:dataTypeARINC429 ObjectUnionOf
25 (fbw:ARINC429Datafield fbw:ARINC429Message))
26 DataPropertyDomain(fbw:dataTypeARINC429 ObjectSomeValuesFrom
27 (fbw:hasARINC429Datafield fbw:ARINC429Datafield))
28 DataPropertyDomain(fbw:dataTypeARINC429 DataSomeValuesFrom
29 (fbw:dataTypeARINC429 xsd:string))
30 DataPropertyDomain(fbw:dataTypeARINC429 DataExactCardinality
31 (1 fbw:dataTypeARINC429 xsd:string))
32 DataPropertyRange(fbw:dataTypeARINC429 xsd:string)
33
34 ##############################
35 #    Classes
36 ##############################
37
38 # Class: fbw:ARINC429 (ARINC 429)
39
40 AnnotationAssertion(rdfs:comment fbw:ARINC429 "The  ARINC 429 Specification defines
41 the standard requirements for the transfer of digital data between
42  avionics systems on commercial aircraft. ARINC 429 is also known as the Mark 33 DITS Specification."@en)
43 AnnotationAssertion(rdfs:label fbw:ARINC429 "ARINC 429"@en)
44 EquivalentClasses(fbw:ARINC429 ObjectIntersectionOf(fbw:ProtocolAvionicsBusStandard ObjectSomeValuesFrom(
       fbw:sendsARINC429Message fbw:ARINC429Message)))
45 SubClassOf(fbw:ARINC429 fbw:ProtocolAvionicsBusStandard)
46 SubClassOf(fbw:ARINC429 ObjectUnionOf(fbw:AFDX fbw:ARINC429))
47 SubClassOf(fbw:ARINC429 ObjectSomeValuesFrom
48 (fbw:sendsARINC429Message fbw:ARINC429Message))
49
50 # Class: fbw:ARINC429Message (ARINC 429 Message)
51
52 AnnotationAssertion(rdfs:comment fbw:ARINC429Message "Most ARINC messages contain only
53 one data word consisting of either Binary (BNR), Binary Coded Decimal (BCD) or
54 Discrete Data Combination of BNR, BCD or individual bit representation."@en)
55 AnnotationAssertion(rdfs:label fbw:ARINC429Message "ARINC 429 Message"@en)
56 EquivalentClasses(fbw:ARINC429Message ObjectIntersectionOf(fbw:ARINC429Message DataSomeValuesFrom(fbw:
       dataTypeARINC429 xsd:string)))
57 SubClassOf(fbw:ARINC429Message fbw:AvionicsBusMessage)
58 SubClassOf(fbw:ARINC429Message ObjectSomeValuesFrom
59 (fbw:hasARINC429Datafield fbw:ARINC429Datafield))
60 SubClassOf(fbw:ARINC429Message DataSomeValuesFrom(fbw:dataTypeARINC429 xsd:string))
61 SubClassOf(fbw:ARINC429Message DataExactCardinality(1 fbw:dataTypeARINC429 xsd:string))
62
63 # Class: fbw:ARINC429MessageBCD (fbw:ARINC429MessageBCD)
64
65 AnnotationAssertion(rdfs:comment fbw:ARINC429MessageBCD "ARINC429 message of type BCD"@eng)
```

```
66 EquivalentClasses(fbw:ARINC429MessageBCD ObjectIntersectionOf(fbw:ARINC429Message DataHasValue(fbw:
       dataTypeARINC429 "BCD")))
67 SubClassOf(fbw:ARINC429MessageBCD fbw:ARINC429Message)
68 SubClassOf(fbw:ARINC429MessageBCD ObjectIntersectionOf(fbw:ARINC429Message DataSomeValuesFrom(fbw:
       dataTypeARINC429 xsd:string)))
69 SubClassOf(fbw:ARINC429MessageBCD ObjectIntersectionOf(fbw:ARINC429Message DataHasValue(fbw:
       dataTypeARINC429 "BCD")))
70 SubClassOf(fbw:ARINC429MessageBCD ObjectUnionOf(fbw:ARINC429MessageBCD fbw:ARINC429MessageBNR fbw:
       ARINC429MessageDiscrete))
71 SubClassOf(fbw:ARINC429MessageBCD ObjectSomeValuesFrom(fbw:hasARINC429Datafield fbw:ARINC429Datafield))
72 SubClassOf(fbw:ARINC429MessageBCD DataHasValue(fbw:dataTypeARINC429 "BCD"))
73 SubClassOf(fbw:ARINC429MessageBCD DataExactCardinality(1 fbw:dataTypeARINC429 xsd:string))
74
75 # Class: fbw:SSMNegativeDataValue (fbw:SSMNegativeDataValue)
76
77 AnnotationAssertionrdfs:comment fbw:SSMNegativeDataValue A BCD label with the value 11 T indicate the word
       s data contains a "negative value" corresponding to "Minus, South, West, Left, From, Below @en
78 EquivalentClasses(fbw:SSMNegativeDataValue ObjectIntersectionOf(fbw:SSM ObjectIntersectionOf(
       ObjectSomeValuesFrom(fbw:isARINC429MessageOf fbw:ARINC429MessageBCD) DataHasValue(fbw:datafieldValue "
       3"^^xsd:integer))))
79 SubClassOf(fbw:SSMNegativeDataValue fbw:SSM)
80 SubClassOf(fbw:SSMNegativeDataValue ObjectIntersectionOf(ObjectSomeValuesFrom(fbw:isARINC429MessageOf fbw:
       ARINC429MessageBCD) DataHasValue(fbw:datafieldValue "3"^^xsd:integer)))
81 SubClassOf(fbw:SSMNegativeDataValue ObjectSomeValuesFrom(fbw:isARINC429MessageOf fbw:ARINC429MessageBCD))
82 SubClassOf(fbw:SSMNegativeDataValue DataSomeValuesFrom(fbw:datafieldValue DatatypeRestriction(xsd:integer
       xsd:minInclusive "0"^^xsd:integer xsd:maxInclusive "3"^^xsd:integer)))
83 SubClassOf(fbw:SSMNegativeDataValue DataHasValue(fbw:LSB "30"^^xsd:integer))
84 SubClassOf(fbw:SSMNegativeDataValue DataHasValue(fbw:MSB "31"^^xsd:integer))
85 SubClassOf(fbw:SSMNegativeDataValue DataHasValue(fbw:datafieldValue "3"^^xsd:integer))
86
87 # Class: fbw:SSMPositiveDataValue (fbw:SSMPositiveDataValue)
88
89 AnnotationAssertion(rdfs:comment fbw:SSMPositiveDataValue A BCD label with the value 00  indicates the word
       s data contains a "positive value" corresponding to "Plus, North, East, Right, To, Above""@en)
90 EquivalentClasses(fbw:SSMPositiveDataValue ObjectIntersectionOf(fbw:SSM ObjectIntersectionOf(
       ObjectAllValuesFrom(fbw:isARINC429MessageOf fbw:ARINC429MessageBCD) DataHasValue(fbw:datafieldValue "0
       "^^xsd:integer))))
91 SubClassOf(fbw:SSMPositiveDataValue fbw:SSM)
92 SubClassOf(fbw:SSMPositiveDataValue
93 ObjectIntersectionOf(ObjectAllValuesFrom(fbw:isARINC429MessageOf fbw:ARINC429MessageBCD) DataHasValue(fbw:
       datafieldValue "0"^^xsd:integer)))
94 SubClassOf(fbw:SSMPositiveDataValue ObjectAllValuesFrom(fbw:isARINC429MessageOf fbw:ARINC429MessageBCD))
95 SubClassOf(fbw:SSMPositiveDataValue DataSomeValuesFrom(fbw:datafieldValue DatatypeRestriction(xsd:integer
       xsd:minInclusive "0"^^xsd:integer xsd:maxInclusive "3"^^xsd:integer)))
96 SubClassOf(fbw:SSMPositiveDataValue DataHasValue(fbw:LSB "30"^^xsd:integer))
97 SubClassOf(fbw:SSMPositiveDataValue DataHasValue(fbw:MSB "31"^^xsd:integer))
98 SubClassOf(fbw:SSMPositiveDataValue DataHasValue(fbw:datafieldValue "0"^^xsd:integer))
```

Listing C.1 Syntaxe fonctionnelle OWL des valeurs «positives» et «négatives» d'un mot ARINC429 de type BCD

```
1
2 Declaration(Class(fbw:FlyByWireFix))
3 Declaration(ObjectProperty(fbw:hasCommunicationFix))
4
5 ##############################
6 #    Classes
7 ##############################
8
9 # Class: fbw:FlyByWireFix (fbw:FlyByWireFix)
10
11 AnnotationAssertion(rdfs:comment fbw:FlyByWireFix "A communication fix defined by Fly-By-Wire coordinates."
       @en)
12 EquivalentClasses(fbw:FlyByWireFix ObjectSomeValuesFrom(fbw:sendsFlyByWireMessage fbw:AvionicsBusMessage))
13 SubClassOf(fbw:FlyByWireFix ObjectSomeValuesFrom(fbw:sendsFlyByWireMessage fbw:AvionicsBusMessage))
14
15 ##############################
16 #    Properties
17 ##############################
18
19 # Object Property: fbw:hasCommunicationFix (fbw:hasCommunicationFix)
20
21 SubObjectPropertyOf(fbw:hasCommunicationFix fbw:aRINCONTOtopObjectProperty)
```

```
AsymmetricObjectProperty(fbw:hasCommunicationFix)
IrreflexiveObjectProperty(fbw:hasCommunicationFix)
ObjectPropertyDomain(fbw:hasCommunicationFix atm:AircraftTrackPoint)
ObjectPropertyRange(fbw:hasCommunicationFix fbw:FlyByWireFix)
```

Listing C.2 Syntaxe fonctionnelle OWL de la relation d'un message ARINC 429 avec le point de piste d'un vol

```

Declaration(Class(fbw:FlyByWireFix))
Declaration(ObjectProperty(d3fend:first-ATTACK-scenario))
Declaration(ObjectProperty(d3fend:next-ATTACK-scenario))

##############################
#    Classes
##############################

# Class: d3fend:ScenarioATTACK (Scenario ATTACK Technique)

AnnotationAssertion(rdfs:comment d3fend:ScenarioATTACK "An attack scenario is composed
 of a collection of Attack Technique"@eng)
AnnotationAssertion(rdfs:label d3fend:ScenarioATTACK "Scenario ATTACK Technique"@eng)
EquivalentClasses(d3fend:ScenarioATTACK ObjectIntersectionOf
  (d3fend:ATTACKThing DataSomeValuesFrom(d3fend:scenarioDescription xsd:string)))
SubClassOf(d3fend:ScenarioATTACK d3fend:ATTACKThing)

##############################
#    Properties
##############################

# Object Property: d3fend:d3fend-scenario-object-property (d3fend-scenario-object-property)

AnnotationAssertion(rdfs:label d3fend:d3fend-scenario-object-property
  "d3fend-scenario-object-property"@eng)
SubObjectPropertyOf(d3fend:d3fend-scenario-object-property d3fend:d3fend-object-property)

# Object Property: d3fend:first-ATTACK-scenario (first-ATTACK-scenario)

AnnotationAssertion(rdfs:label d3fend:first-ATTACK-scenario "first-ATTACK-scenario"@eng)
SubObjectPropertyOf(d3fend:first-ATTACK-scenario d3fend:d3fend-scenario-object-property)
FunctionalObjectProperty(d3fend:first-ATTACK-scenario)
AsymmetricObjectProperty(d3fend:first-ATTACK-scenario)
IrreflexiveObjectProperty(d3fend:first-ATTACK-scenario)
ObjectPropertyDomain(d3fend:first-ATTACK-scenario d3fend:ScenarioATTACK)
ObjectPropertyRange(d3fend:first-ATTACK-scenario d3fend:OffensiveTechnique)

# Object Property: d3fend:next-ATTACK-scenario (next-ATTACK-scenario)

AnnotationAssertion(rdfs:comment d3fend:next-ATTACK-scenario "an ATTACK
 Technique from a scenario following another one"@eng)
AnnotationAssertion(rdfs:label d3fend:next-ATTACK-scenario "next-ATTACK-scenario")
SubObjectPropertyOf(d3fend:next-ATTACK-scenario d3fend:d3fend-scenario-object-property)
FunctionalObjectProperty(d3fend:next-ATTACK-scenario)
IrreflexiveObjectProperty(d3fend:next-ATTACK-scenario)
ObjectPropertyDomain(d3fend:next-ATTACK-scenario d3fend:OffensiveTechnique)
ObjectPropertyRange(d3fend:next-ATTACK-scenario d3fend:OffensiveTechnique)
```

Listing C.3 Syntaxe fonctionnelle OWL d'un scénario d'attaque