Institut pour la Maîtrise des Risques
Sûreté de Fonctionnement - Management - Cindyniques

Utilisation des modèles d'ingénierie système de Capella/ARCADIA pour l'application de la méthode System-Theoretic Process Analysis (STPA)

Using System Engineering Model from Capella/ARCADIA to apply System-Theoretic Process Analysis (STPA)

VIARDOT Jean-François
LGM
Vélizy-Villacoublay, France
jean-francois.viardot@lgm.fr

MONTIGAUD Thibault
LGM
Vélizy-Villacoublay, France
thibault.montigaud@lgm.fr

I. RÉSUMÉS

Résumé — STPA (System-Theoretic Process Analysis) est une méthode d'analyse du risque qui se base sur le cadre méthodologique STAMP (System-Theoretic Accident Model and Processes) et son modèle de boucle de contrôle. Cette méthode devient de plus en plus présente notamment dans les phases amont du cycle en V. STAMP s'illustrant par sa vision axée sur la théorie des systèmes, la méthode est ainsi en lien étroit avec l'ingénierie système bien que ce cadre propose son propre modèle différent du MBSE (Model-Based System Engineering). Afin de réduire le temps de réalisation de l'étude STPA et d'assurer une cohérence entre les ingénieurs système et les ingénieurs sûreté de fonctionnement, ce papier propose une manière d'exploiter les modèles MBSE fournit par la méthodologie ARCADIA pour l'application de STPA.

Ainsi dans cet article est décrit une méthode permettant de passer d'un modèle ARCADIA réalisé sur Capella a un modèle reprenant les caractéristiques de STAMP. Grâce à l'exploitation des modèles MBSE, la phase d'identification des interactions de contrôle gagne un aspect plus systématique qui permet un gain de temps dans l'application de la méthode STPA.

Mots-clefs — STPA, STAMP, ARCADIA, Capella, Model-Based Engineering.

Abstact — System-Theoretic Process Analysis (STPA) is a risk analysis method based on System-Theoretic Accident Model and Processes (STAMP) and its control loop model. Its usage is growing up, especially in the early phases of the V-model. As STAMP is entrenched in System Theory, it is embedded with system engineering. However, it has its own framework and model that differ from those provided by Model-Based System Engineering (MBSE). The purpose of this paper is to propose a method that makes consistency between safety and system engineering easier. It will therefore make the STPA process more efficient by reducing the overall analysis time.

For that matter, herein a process that translate a Capella model based on ARCADIA to a STAMP-like model is defined by operating the fewer modifications to the original. Thanks to the use of the MBSE, the control actions identification process is made easier.

Keywords — STPA, STAMP, ARCADIA, Capella, Model-Based Engineering.

II. Introduction

STPA, méthode conçue par les chercheurs du MIT, Nancy Leveson et John Thomas **(N. G. Leveson & Thomas, 2018)**, a pour but de rassembler l'ingénierie système et la sûreté de fonctionnement autour de concepts et de modèles unifiés afin d'intégrer dès la phase de conception du système les éléments liés à la maitrise des risques. La méthode ayant fait l'objet d'un fort lobbying de la part du M.I.T, a commencé à s'installer dans le paysage européen. C'est dans ce cadre notamment que différentes études et mise en place sur des cas d'usage ont pu être partagé **(Thibault et al., 2022)**. Ceux-ci présentaient régulièrement en donnée d'entrée des modèles MBSE suivant la méthodologie ARCADIA, décrite dans le guide pratique **(Roques, 2017)**, réalisés par des équipes d'ingénierie système. Les principes du MBSE et d'ARCADIA ne seront pas rappelés aux lecteurs dans ce document. Dès lors, l'intérêt était alors d'établir la manière de lier ces modèles et celui proposé par STAMP. Bien que plusieurs articles fassent état de l'utilisation de SysML pour la modélisation STAMP/STPA **(de Souza et al., 2020)**, ils restaient au niveau de la modélisation SysML, sans considérer une méthodologie structurante de MBSE. La méthodologie ARCADIA, par son organisation et sa structuration des données contenues dans les diagrammes SysML, vient apporter un aspect systématique à cette transition. Notre choix de l'outil Capella s'explique principalement par le fait que cet outil est open source, son usage est largement répandu aussi bien au sein de THALES, qui est le partenaire historique de LGM autour de la R&D (Recherche & Développement) de la méthodologie STPA, que pour de nombreux autres industriels. L'accessibilité de l'outil a permis la réalisation de nombreux cas d'utilisation qui sont venu étoffer la méthode développée ici. Des considérations similaires pourraient être menées au travers des modèles construits au sein de l'outil Cameo System Modeler ou Rational Rhapsody. Cette présentation dresse une procédure permettant l'association des éléments constitutifs des modèles Capella/ARCADIA et de STAMP. La démarche méthodologique est présentée ici en rappelant dans un premier temps les notions fondamentales de STPA. Des extraits d'un cas d'application sont également partagés.

III. System-Theoretic Process Analysis

Dans un premier temps, il est nécessaire de rappeler quelques concepts fondamentaux et points de vocabulaire de STAMP/STPA. Les termes utilisés se rapportant à la méthode seront conservés dans leur langue d'origine, l'anglais, afin de conserver leur sémantique. Chacun sera définit suivant le guide du M.I.T (N. G. Leveson & Thomas, 2018)

La méthode STPA se construit autour de trois grandes phases :

- La foundation phase : On y définit, avec les différentes parties prenantes du système, ses limites ainsi que ses « **losses »**. Grâce à ceux-ci, il est possible de définir les **« hazards »** qui nous guide dans la construction des « **system-level constraints** » ;
- La modelling phase : On y construit la **« Hierarchical Control Structure »** constituée d'une ou plusieurs « **Control Loop ».** De celle-ci, il est possible de tirer les différentes « **Control Action »** qui existent entre chaque constituant du système. Cette phase se base sur la réalisation du modèle associé au cadre méthodologie STAMP comme définit par Nancy Leveson (N. Leveson, 2011)
- L'analysing phase : On recherche les « Control Actions » qui peuvent par leur comportement mener à un **«** hazard », les UCA (Unsafe Control Actions)

Loss: A loss involves something of value to stakeholders. Losses may include a loss of human life or human injury, property damage, environmental pollution, loss of mission, loss of reputation, loss or leak of sensitive information, or any other loss that is unacceptable to the stakeholders.

Il s'agit d'enjeux pour les parties prenantes du système, d'évènements jugés inacceptable pour eux.

Hazard: system state or set of conditions that, together with a particular set of worst-case environmental conditions, will lead to a loss.

Control Loop: In general, a controller may provide control actions to control some process and to enforce constraints on the behavior of the controlled process. The control algorithm represents the controller's decision-making process—it determines the control actions to provide. Controllers also have process models that represent the controller's internal beliefs used to make decisions. Process models may include beliefs about the process being controlled or other relevant aspects of the system or the environment. (Voir **Erreur ! Source du renvoi introuvable.**)

Hierarchical Control Structure: system model that is composed of feedback control loops. An effective control structure will enforce constraints on the behavior of the overall system.

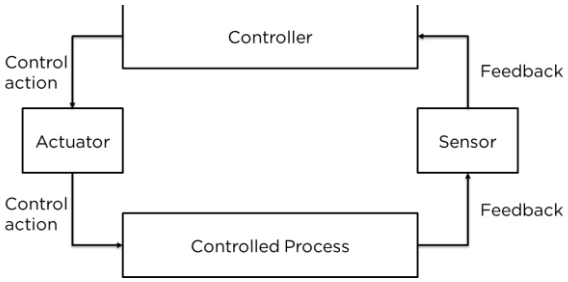

Figure 1 Modèle générique simplifié d'une control loop

IV. METHODOLOGIE

Les éléments de contextes autour de STAMP/STPA ayant été rappelés plus haut. Ce paragraphe présente comment la méthodologie a été établie, détaille sa mise en œuvre et propose des exemples appliqués à un cas d'usage open source. La démarche scientifique mise en place pour l'élaboration de cette méthodologie est la suivante :

- Analyse comparative du MBSE et de la modélisation en *Hierarchical Control Structure*,
- Etablissement d'une méthodologie a priori,
- Mise en œuvre sur des cas d'usages et cas tests,
- Correction par itération de la méthodologie et analyse des limites restantes.

La suite de cette présentation, fait état de l'étude comparative et de la dernière itération de la méthodologie sans présenter les différentes évolutions.

A. Etude comparative

La méthode ARCADIA se construit autour de quatre principaux niveaux d'analyse dont les définitions accessibles directement dans l'outil Capella sont :

- L'analyse opérationnelle : Analyse des besoins et de l'environnement des stakeholders. Identifier les entités, acteurs, rôles, activités, concepts ;
- L'analyse système : Formaliser et consolider les exigences système, identifier ses limites ;
- L'architecture logique : Définir le fonctionnement du système afin de répondre aux attentes ;
- L'architecture physique : Définir comment le système sera développé et construit.

La méthode STPA, quant à elle, présente les étapes suivantes définies dans l'article (Thibault et al., 2022)

- Définir les objectifs de l'analyse : Le périmètre de l'analyse, les *losses* et les *hazards*, et fournir des exigences de sécurité au niveau système ;
- Etablir les structures de contrôle ;
- Identifier les UCA à partir des structures de contrôle et définir des exigences de sécurité sur le comportement attendu des différents contrôleurs ;
- Identifier les scénarios conduisant aux *losses* et définir des exigences de sécurité applicable sur tous les éléments contribuant à la réalisation d'une ou plusieurs fonctionnalités.

Le lien entre ces deux méthodes s'illustre dans les deux premières étapes de STPA, car l'objectif ici est de pouvoir dériver les structures de contrôle depuis ARCADIA. Ainsi, il est possible d'inclure la phase de fondation au travers des modèles ARCADIA comme présenté dans la figure ci-dessous :

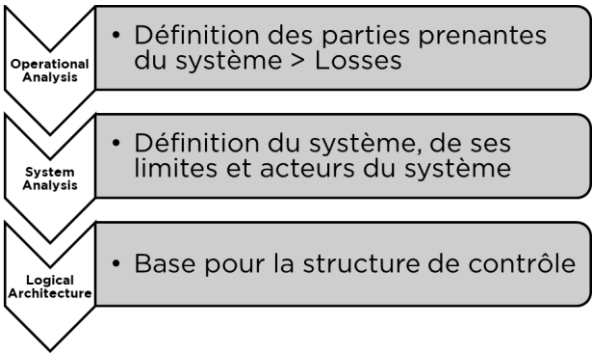

Figure 2 Correspondance ARCADIA/STPA

Dans un second temps, le but était de définir les différents éléments du modèle de boucle de contrôle afin d'observer des similitudes avec les artéfacts ARCADIA. Si l'on revient sur le modèle générique présenté en Figure 1, une boucle de contrôle se constitue de plusieurs éléments :

- Les *controllers* qui sont des entités pouvant être aussi bien des sous-systèmes que des acteurs ou organisme intervenant dans les fonctions du systèmes étudiés. Ceux-ci vont communiquer différents ordres, commandes et informations entre eux ;
- Les *actuators,* sont une typologie de contrôleur qui ne font que transmettre les actions de contrôle entre les différents contrôleurs ;
- Les *sensors,* eux, transmettent uniquement des informations (*feedback*) entre les différents contrôleurs ;
- Les *control actions* sont les flux de commandes qu'échangent les différents contrôleurs entre eux ou avec le processus contrôlé ;
- Les *feedbacks* sont les flux d'informations qu'échangent les différents contrôleurs entre eux ou le processus contrôlé avec les contrôleurs.

En observant ces différents concepts, on remarque des similarités avec des artéfacts Capella. En effet, les *controllers, actuators* et *sensors* étant, comme mentionné plus tôt, des entités agissant au sein du système ou avec celui-ci peuvent se comparer à des acteurs, entités ou composants dans Capella. La méthodologie STAMP regroupant aussi bien les acteurs, organismes ou sous-systèmes sous la même étiquette de contrôleur, tous ces éléments des modèles ARCADIA peuvent être considérés.

La notion de flux de commande et d'information quant à elle intervient principalement au travers des architectures logiques d'ARCADIA. Le focus a donc été fait sur les modèles de cette couche où les échanges fonctionnels ainsi que les fonctions elles-mêmes fournissent des données sur la manière dont chaque composant logique ou acteur interagit avec les autres. Ceux-ci ont donc ainsi été identifiés aux *control actions* quand l'on communiquait des ordres ou commandes, et des *feedbacks* quand il s'agissait exclusivement d'informations.

Afin de différencier les *controllers, actuators* et *sensors,* le focus s'est fait autour du sens des échanges. Contrairement aux *controllers*, les *actuators* et *sensors* réalisent des échanges unilatéraux. L'objectif est donc d'identifier, dans les modèles ARCADIA, les composants logiques qui réalisent des échanges fonctionnels unilatéraux et d'observer de quel type d'échange il s'agit. Si ce sont exclusivement des commandes ou des ordres, on rapprochera ces composants aux *actuators*. A l'inverse si ce sont exclusivement des échanges de données et autres informations, on les rapprochera des *sensors.*

Pour valider la pertinence de la corrélation ARCADIA/STAMP proposée, plusieurs cas d'application ont été réalisés. Un d'entre eux servira de fil conducteur à l'illustration de la méthodologie proposée au sein de cette publication.

B. Méthodologie illustrée sur un cas d'application

Le cas d'étude proposé est celui du système AIDA (Aircraft Inspection by Drone Assistant) de l'IRT Saint-Exupéry accessible à tous. L'ensemble des modèles Capella ayant été réalisés dans le cadre du projet S2C (IRT, 2022) sont mis à disposition, cependant il n'est pas nécessaire de posséder autant de modèles ou de diagrammes pour mettre en place la méthodologie, seule une partie présentée à la suite a été retenue.

Le système AIDA est une assistance aux procédures de vérification pré-vol par drone pour l'aviation civile. Les **Erreur ! Source du renvoi introuvable.** et **Erreur ! Source du renvoi introuvable.** présentent deux des modèles Capella du système qui ont servi à ce cas d'étude.

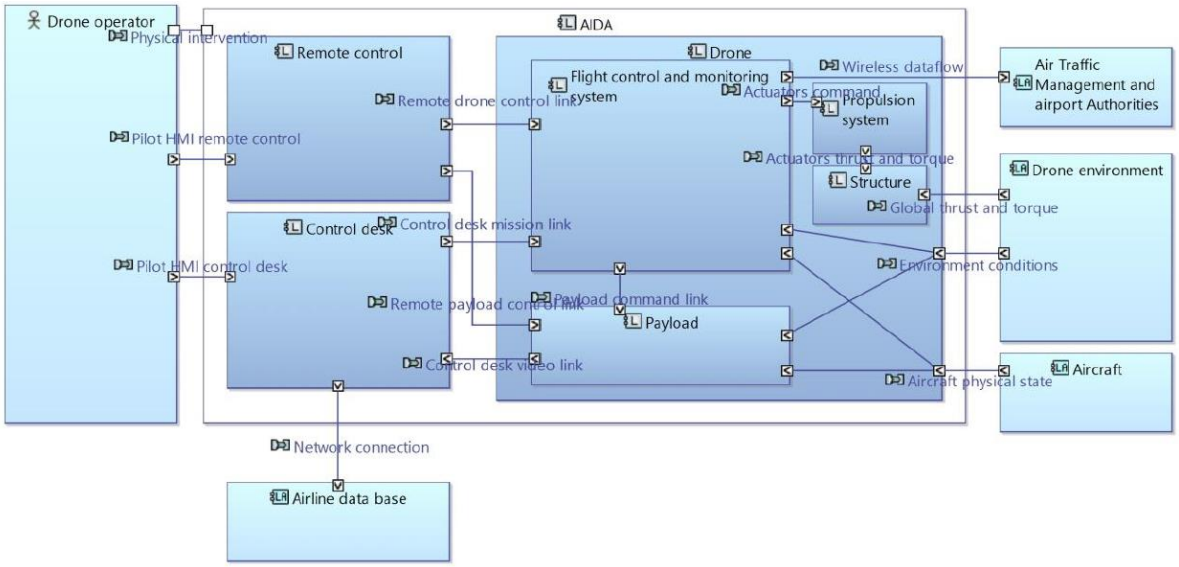

Figure 3 Architecture logique haut niveau de AIDA

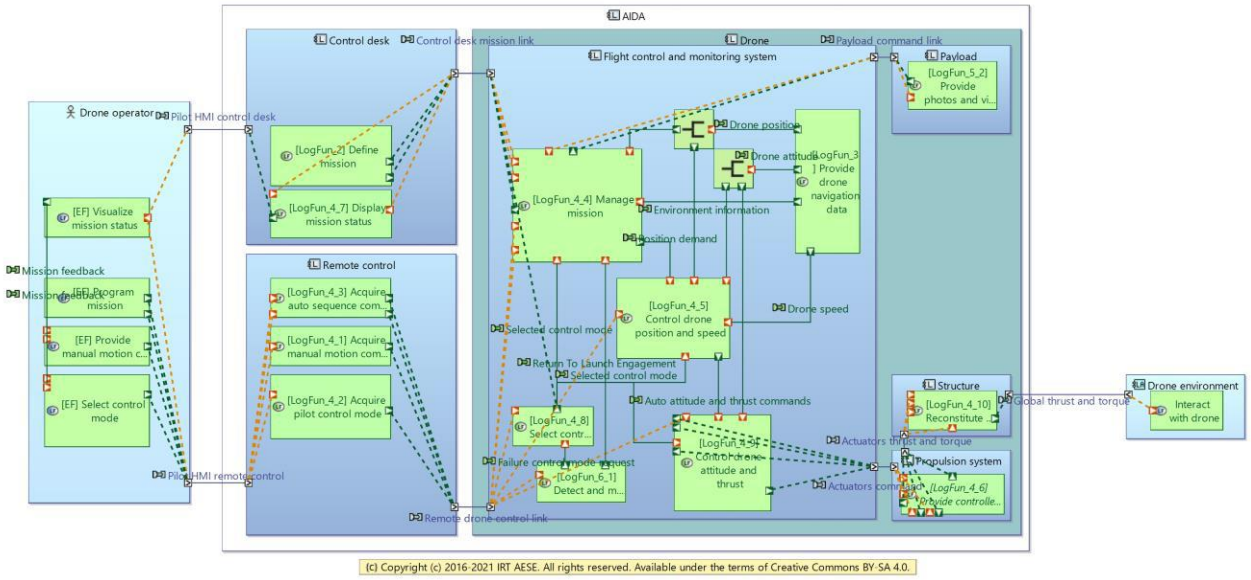

Figure 4 Contextual Logical Architecture of function Control drone Motion

1) Définition des objectifs de l'analyse (Losses – Hazards – System-level constraints)

Au sein d'ARCADIA, les parties prenantes du système sont recensées en tant qu'entité ou acteur opérationnel de l'architecture opérationnelle (**Erreur ! Source du renvoi introuvable.**) au travers par exemple du diagramme des parties prenantes.

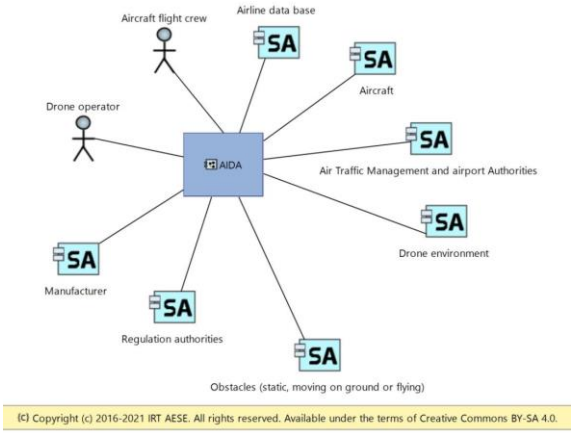

Figure 5 AIDA Stakeholders diagram

Les différentes « capabilities » de haut niveau, exprimées au sein de l'analyse opérationnelle, permettent également de faire ressortir les attentes fortes et enjeux de niveau système.

Grâce à la prise en compte de ces éléments, il est possible d'identifier les *losses*. Dans notre cas d'étude, cela donne par exemple les éléments suivants :

- L-1 : Incapacité du drone à accomplir la mission
- L-2 : Blessure d'un opérateur
- L-3 : Perte de contrôle du drone
- L-4 : Dégradation de l'avion inspecté

La liste des parties prenantes devient alors évidente depuis ARCADIA et le diagramme des parties prenantes.

Il reste toutefois une partie de cette phase de définition des objectifs pour STAMP/STPA qui n'est pas directement accessible depuis les modèles Capella, c'est la définition des *Hazards*. En effet, c'est une phase analytique qui demande de se questionner sur les différentes situations qui pourraient placer notre système aux abords d'un ou plusieurs *losses*. Cette phase doit malheureusement toujours être traité en dehors et cela se fait généralement par des séances de travail avec une équipe pluridisciplinaire animée par un ingénieur sûreté de fonctionnement. Cette approche est similaire à celles d'identification des risques, bien qu'ici avec un formalisme différent est un positionnement en situation dangereuse et non en « événement redouté ».

Une fois, les *hazards* définies et l'établissement des exigences de sécurité de niveau système, vient la phase de modélisation.

2) Etablissement des strcutures de contrôle hiérarchiques

C'est dans cette phase que l'architecture système et logique d'ARCADIA ont le plus d'apport. En effet, ces deux niveaux permettent de représenter les interactions externes et internes du système. Grâce à cela on peut entrevoir quels sont les parties et acteurs du système et de quelle manière ils contrôlent ou reçoivent les flux d'information à leur disposition.

Si l'on met les deux structures en parallèle, il est possible d'observer des similarités qui nous permettent de corréler les artéfacts Capella organisé par l'approche ARCADIA avec ceux de STAMP.

Les différents acteurs et composants des architectures ARCADIA envoient et reçoivent des flux de commande et de message (*Component Exchange*). Dès lors il est possible de les assimiler à des contrôleurs et leurs flux à des actions de contrôle ou des *feedbacks*.

Afin de différencier les types de flux, il faut identifier le niveau d'autorité des différents contrôleurs du système. Nous avons pu établir dans un premier temps que les composants et acteurs qui communiquent exclusivement des flux de commande comme, dans le cas d'AIDA, le « propulsion system » sont des « actuators ». A l'inverse, ceux qui ne communiquent que des flux de messages/données sont des « sensors ». La hiérarchie des contrôleurs se définit par la suite en observant l'ensemble des flux de commande. Le contrôleur de plus haut niveau envoie des commandes, mais ne reçoit que des messages/données, ici le « drone operator », le reste de la hiérarchie se construit en suivant l'ordre des commandes en partant de ce contrôleur.

Selon la formulation des échanges fonctionnels et des fonctions, il peut-être plus judicieux pour comprendre les actions de certains contrôleurs, d'identifier les fonctions comme des flux de commande également et donc des control actions. C'est le cas de la fonction « select control mode », qui vient apporter une information supplémentaire quant au flux de commande initial « pilot HMI remote control ».

Enfin le *Controlled Process* reste un élément plus abstrait. Celui-ci représente la mission adjointe au système. On peut ainsi le retrouver dans l'analyse opérationnelle comme étant la *Main Mission* dans les « Missions and Capabilites Diagrams ».

Une fois chaque élément identifié, il est possible de simplement les réorganiser afin de revenir au format classique de la *Hierarchical Control Structure*. On organise pour cela les contrôleurs verticalement suivant leur hiérarchie en regroupant les actions de contrôle et *feedbacks* identifiés dans les différentes architectures Capella. On obtient ainsi le modèle en **Erreur ! Source du renvoi introuvable.** dans lequel on peut dresser les correspondances suivantes :

Elément du modèle STAMP	**Origine Capella/ARCADIA**
Controller: Drone Operator	[LAB] Logical Actor
Controller: Control desk	[LAB] Logical Component
Controller: Flight Control and Monitoring System (FCMS)	[LAB] Logical Component
Actuator: Remote Control	[LAB] Logical component
Actuator: Propulsion system	[LAB] Logical component
Sensor: Payload	[LAB] Logical component
Control Action: Pilot HMI control desk	[LAB] Component Exchange
Control Action: Select control mode	[LAB] Function
Control Action: Define mission	[LAB] Function
Control Action: Actuators command	[LAB] Component Exchange
Feedback: Drone position	[LAB] Functional Exchange
Feedback: Drone attitude	[LAB] Functional Exchange
Feedback: Photos and videos	[LAB] Function (Provide photos and videos)
Controlled Process: Realise the Pre-Flight Check	[MB] Mission

Tableau 1 Association des artéfacts STAMP et Capella pour AIDA

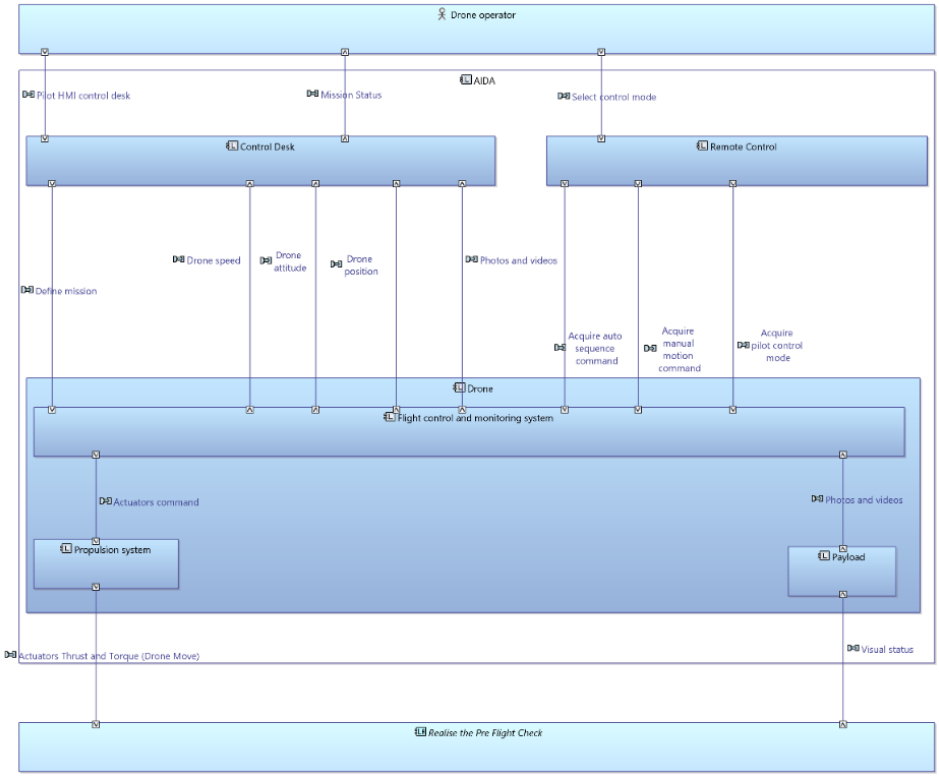

Figure 6 AIDA Hierarchical Control Structure

C. *Synthèse de l'équivalence des artéfacts*

Des différentes expérimentations nous avons pu établir les correspondances suivantes pour assurer la transition du modèle Capella/ARCADIA vers la *Hierarchical Control Structure* de STAMP :

STAMP	**Capella/ARCADIA**
Control Action	Component Exchange and function (Partant d'un contrôleur de haute autorité vers un contrôleur de plus faible autorité ou vers le controlled process)
Feedback	Component Exchange (Partant du controlled process ou d'un contrôleur d'autorité faible vers un contrôler d'autorité plus haute)
Controller	Actor/Entity et component
Controlled Process	Main mission
Actuator	Actor/entity et component (où l'ensemble des échanges est reçu d'un contrôleur de haute autorité et envoyé vers un contrôleur de plus faible autorité)
Sensor	Actor/entity et component (où l'ensemble des échanges est reçu d'une faible autorité et envoyé vers une plus haute)

Tableau 2 Equivalence des artéfacts STAMP et Capella

V. LIMITATIONS

Si l'utilisation de ces modèles permet de générer la majorité de la phase de définition des objectifs de STPA, certains éléments restent toutefois dépendant de la réflexion de l'ingénieur. C'est le cas de la définition des *hazards* qui sont un élément charnière pour la construction du reste de l'étude. Si cela peut sembler être un frein quant à l'optimisation du temps d'application de la méthodologie au premier abord, cela est perçu par les auteurs et leurs expériences de la méthodologie STPA comme un point important et bénéfique. En effet, la définition des *hazards* et des exigences de sécurité de niveau système est primordiale dans l'approche STPA et permet généralement d'apporter des orientations bénéfiques aux architectures, à condition que ces éléments soient partagés et définis en co-ingénierie avec les architectes système. Ce travail ne doit pas être automatisé, mais doit au contraire donné lieux à des échanges ouverts qui permettront de définir des pistes de solutions et des contraintes importantes pour la sécurité des systèmes.

De plus, la procédure qui a été définie demande un filtrage non systématique des données des modèles d'ARCADIA afin de définir les actions de contrôle. En effet, celle-ci peuvent s'illustrer au travers de plusieurs éléments d'ARCADIA, notamment des échanges fonctionnels ou des fonctions elle-même selon leur définition dans le modèle Capella. Ce besoin de filtrer les informations se constate également au niveau des différents modèles Capella. Il appartient à celui qui réalise l'étude de définir les informations utiles parmi les architectures disponibles.

CONCLUSION

La méthodologie STPA est un vecteur intéressant pour faciliter la communication entre les deux métiers que sont l'ingénierie système et la sûreté de fonctionnement, il ne faut donc pas que la réalisation des modèle STAMP soit un frein. Il est nécessaire de proposer des solutions permettant de concilier MBSE et STAMP. Les deux catégories de modèle, possédant de nombreuses similarités, il est possible d'élaborer des passerelles depuis le MBSE avec ARCADIA vers les modèles STAMP. Ce papier propose une approche méthodologique permettant de définir cette transformation, bien que quelques difficultés restent à traiter. On pourrait ainsi envisager, dans un avenir proche, que Capella effectue directement la transformation de ses différents modèles en modèle STAMP. Toutefois, le besoin de filtrer les différents modèles ARCADIA ainsi que la qualité des modèles initiaux pourrait être un obstacle à cette automatisation de l'approche.

REMERCIEMENTS

Les auteurs souhaitent remercier les intervenants de LGM qui ont pris le temps de contribuer à ce sujet ou qui ont enrichi les débats en confrontant des cas d'usage. Un remerciement particulier est également adressé aux personnes de THALES qui ont réalisé des travaux sur des sujets connexes et qui ont permis de mettre en perspective ces travaux à l'aune de leurs expériences.

REFERENCES

de Souza, F. G. R., de Melo Bezerra, J., Hirata, C. M., de Saqui-Sannes, P., & Apvrille, L. (2020). Combining STPA with SysML Modeling. *2020 IEEE International Systems Conference (SysCon)*, 1-8. https://doi.org/10.1109/SysCon47679.2020.9275867

IRT, I. A. S.-E. (2022, octobre 21). *AIDA / AIDAArchitecture*. GitLab. https://sahara.irt-saintexupery.com/AIDA/AIDAArchitecture

Leveson, N. (2011). *Engineering a safer world : Systems thinking applied to safety*. MIT Press.

Leveson, N. G., & Thomas, J. P. (2018). STPA handbook. *Cambridge, MA, USA*.

Roques, P. (2017). *Systems Architecture Modeling with the Arcadia Method : A Practical Guide to Capella*. Elsevier.

Thibault, M., Patrice, R., Sébastien, M., Alexandre, T., Gauthier, E., & Perrin de Kierzkowski, J. (2022, octobre). Assessment of STPA methodology and first feedback on use cases. *Congrès Lambda Mu 23 " Innovations et maîtrise des risques pour un avenir durable " - 23e Congrès de Maîtrise des Risques et de Sûreté de Fonctionnement, Institut pour la Maîtrise des Risques*. https://hal.science/hal-03878539