

# MODÉLISATION COMPORTEMENTALE D'UN DATA CENTER POUR LE DÉVELOPPEMENT D'UN JUMENTAUX NUMÉRIQUE

MAROUANE EL MERNISSI<sup>1</sup>, ROLAND DONAT<sup>2</sup>, MOHAMMED HAMZAOU<sup>3</sup>, NATHALIE JULIEN<sup>4</sup>

<sup>1</sup> EdgeMind- UBS  
Vannes, France

[marouane.el-mernissi@univ-ubs.fr](mailto:marouane.el-mernissi@univ-ubs.fr)

<sup>2</sup> EdgeMind  
Vannes, France

[roland.donat@edgemind.net](mailto:roland.donat@edgemind.net)

<sup>3</sup> UBS  
Lorient, France

[mohammed.hamzaoui@univ-ubs.fr](mailto:mohammed.hamzaoui@univ-ubs.fr)

<sup>4</sup> UBS  
Lorient, France

[nathalie.julien@univ-ubs.fr](mailto:nathalie.julien@univ-ubs.fr)

---

**Résumé** – Cet article présente une approche permettant de construire le jumeau numérique d'un data center reposant sur les techniques de modélisation comportementale inspirées de la méthodologie MBSA (Model-Based Safety Assessment). Cette approche consiste dans un premier temps à élaborer une bibliothèque de modélisation générique décrivant les comportements fonctionnels et dysfonctionnels des composants élémentaires participant aux missions d'un data center. Cette bibliothèque est ensuite utilisée pour développer le jumeau numérique d'un data center. Le jumeau est enfin utilisé afin de simuler le système dans le but de prévoir sa performance future et ses éventuels points faibles. Du point de vue technique, le jumeau est développé à partir de l'outil de modélisation et de simulation PyCATSHOO permettant ainsi le calcul de différents indicateurs de fiabilité et de disponibilité, ainsi que l'exploration de scénarios critiques.

**Mots clés** – Jumeau numérique, Modélisation, Maintenance prédictive

---

**Abstract** – This article presents a new approach to build the digital twin of a data center based on behavioral modeling techniques inspired by the MBSA (Model-Based Safety Assessment) methodology. This approach is based on the development of a generic modeling library describing the functional and dysfunctional behaviors of the elementary components involved in the missions of a data center. This library is then used to build the digital twin of a data center. The twin is then used to simulate the system in order to predict its future performance and its possible weak points. From a technical point of view, the twin is developed using the PyCATSHOO modeling and simulation tool, which allows the calculation of various reliability and availability indicators, as well as the exploration of critical scenarios.

**Keywords** – Digital twin, Modeling, Predictive maintenance

---

## 1. INTRODUCTION

Un data center est un lieu qui héberge des équipements informatiques et de stockage de données, qui sont utilisés pour traiter, stocker et diffuser des données pour les entreprises, les gouvernements et d'autres organisations. Les data centers sont soumis à de forts objectifs de disponibilité du fait du caractère critique de l'accès aux données pour la performance et l'organisation des entreprises et institutions.

Pour garantir la disponibilité d'un data center, il est important de mettre en place des mesures de redondance et de reprise après un incident, telles que des serveurs de secours et des systèmes de sauvegarde de données, etc.

Afin de standardiser une échelle de sécurité et de performance chez les exploitants, l'organisme Uptime Institute [Pearl, 2003], a proposé une certification de conception, construction et durabilité opérationnelle pour les data centers dans le monde. Cette certification est déclinée en 4 niveaux, nommés de Tier I à IV, le Tier IV étant le niveau le plus haut, proposant un taux de disponibilité de 99,99% qui est équivalent à moins de 24 minutes d'arrêts cumulés par an grâce à la redondance active des équipements. En France, les certifications correspondantes à ces niveaux de sécurité sont effectuées depuis le mois de janvier 2014.

Pour assurer une telle continuité de service, la maintenance des équipements est primordiale. Bien qu'encore partielle, la maintenance préventive (PM) est parfois négligée en tant qu'outil important de contrôle et d'optimisation.

Toutefois, la maintenance préventive ne suffit pas nécessairement pour garantir une telle disponibilité. Le Tier III ou IV sont caractérisés par une forte redondance des composants et une double source d'alimentation. Cependant, cette multiplication des matériels entraîne davantage de pannes potentielles et rend les interventions de maintenance/dépannage plus complexes [Lebreton, 2016].

Créer une réplique numérique des équipements du data center, appelée également jumeau numérique (JN), a pour objectif de faciliter l'accès à l'information sur le système, permettant *in fine* d'améliorer son pilotage ainsi que son maintien en conditions opérationnelles. Aujourd'hui, le planning de maintenance préventive et les gammes de maintenance de chaque équipement sont intégrés dans un outil de maintenance de type GMAO (gestion de maintenance assistée par ordinateur). Cependant, ces données de GMAO sont loin d'être exploitées à leur plein potentiel.

En croisant les données issues de la GMAO et de l'IoT (Internet of Things), les professionnels de la maintenance sont en mesure de détecter tout écart de fonctionnement et ainsi tenter de

prévenir d'éventuelles anomalies ou pannes.

Un jumeau numérique ou Digital twin est une représentation virtuelle dynamique d'un objet, d'un processus ou d'un service, qui suit son fonctionnement tout au long de son cycle de vie et rend compte des différents événements, changements ou actions qui peuvent survenir dans le temps. Si la technologie a fait son apparition dans l'industrie aérospatiale dans les années 1960, le terme réel de jumeau numérique a été défini pour la première fois en 2002 par Mickael Grieves. Au fil des années, le jumeau numérique a fait son apparition dans divers domaines tels que l'industrie [Liu et al., 2021], le transport [Yan et al., 2021], le bâtiment [Shu et al., 2019], l'énergie [Karanjkar et al., 2018] et la santé [Erol et al., 2020]. Cela a conduit à un intérêt croissant de la part des entreprises et de plusieurs communautés scientifiques, et à une croissance rapide de cette technologie [Blanchet, et al., 2022].

## 2. ÉTAT DE L'ART

Lors de la conception de jumeaux numériques, l'ingénierie système basée sur les modèles (MBSE, *Model-Based Systems Engineering*) et l'évaluation de sécurité basée sur les modèles (MBSA, *Model Based Safety Assessment*) sont deux approches qui jouent un rôle essentiel.

Le MBSE est une approche d'ingénierie qui repose sur l'utilisation de modèles formels et d'outils pour représenter et analyser les systèmes complexes [Estefan, 2008]. Elle vise à améliorer la communication, la coordination et la gestion des projets d'ingénierie en offrant une représentation unifiée et cohérente des différentes facettes d'un système.

L'approche MBSA quant à elle, utilise des modèles de simulation pour évaluer la performance et la sûreté de fonctionnement du système [Li et al., 2014]. Ces modèles peuvent inclure des informations sur les contrôles de sécurité, les mesures de prévention et les procédures d'intervention en cas d'incident. D'un point de vue technique, ils peuvent être utilisés pour prédire comment le système se comportera dans des conditions spécifiques et pour identifier les problèmes potentiels avant la mise en production.

L'intégration de l'ingénierie système basée sur les modèles (MBSE) et de l'évaluation de sécurité basée sur les modèles (MBSA) est une question cruciale pour la conception et l'évaluation des JN destinée au pilotage prévisionnel des performances d'un système. Toutefois, il n'existe pas encore de méthode générale pour créer des modèles intégrant à la fois les considérations MBSA et MBSE [Vidalie et al., 2020].

Les travaux présentés dans cet article proposent une approche originale de construction du modèle comportemental d'un JN en se basant sur une démarche MBSA/MBSE [Demachy et

Guilmeau, 2022] avancée pour représenter et simuler l'exploitation d'un data center en tenant compte des aléas techniques au cours du temps. Des travaux exploratoires similaires ont été menés dans le domaine ferroviaire, où la méthode de modélisation s'appuie sur une présentation du système résultant d'une compréhension partagée entre les ingénieurs FMDS (Fiabilité, Maintenabilité, Disponibilité, Sécurité) et les ingénieurs systèmes, comme décrit dans l'étude de [Donat, 2022]. La démarche de modélisation proposée par les auteurs repose sur les principes suivants :

- Représenter les comportements fonctionnels et dysfonctionnels des différents composants participant aux missions du système étudié.
- Représenter explicitement l'architecture organique du système réel à partir des composants modélisés, à l'inverse des formalismes abstraits classiques tels que les arbres de défaillances ou encore les réseaux de Petri [Collart-Dutilleul, 2009].
- Exploiter le modèle système obtenu grâce à des outils de quantification permettant de réaliser diverses analyses (FMD ou autres).

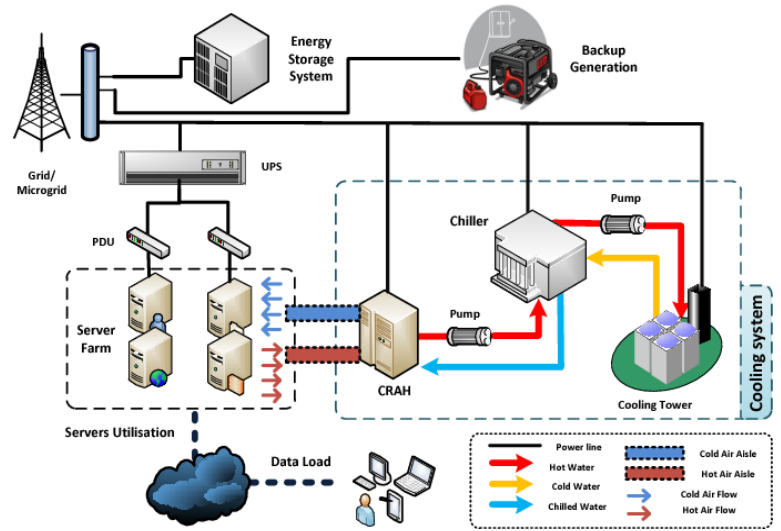
L'approche proposée dans cet article est illustrée via un cas d'étude concret [Rahmani et al., 2018]. La section 3 présente l'architecture générale d'un data center, la section 4 résume les étapes de construction du modèle comportemental ainsi que la présentation des outils de modélisation qui supporte cette approche. La section 5 consiste à exploiter le modèle pour simuler différents scénarios de performance afin d'identifier et anticiper les situations critiques pour l'exploitation. Enfin, la section 6 présente notre retour d'expérience sur la méthodologie de modélisation des JN mise en œuvre en concluant avec quelques perspectives.

### 3. ARCHITECTURE INTERNE D'UN DATA CENTER

Un data center est composé à la fois d'installations industrielles et d'infrastructures informatiques.

Le fonctionnement et la fiabilité d'un tel système nécessitent l'implication de différents composants potentiellement redondants du fait de leur criticité [Douchet, 2015] représentant ainsi des investissements importants.

La Figure 1 illustre les flux de données, d'air et d'électricité, ainsi que l'interconnexion de tous les composants d'un data center typique [Rahmani et al., 2018].



**Figure 1. Architecture générale d'un data center [Rahmani et al., 2018]**

Les sections suivantes présentent les principaux composants d'un data center.

#### 3.1. Chambre des serveurs

Elle est considérée comme la partie principale d'un data center, car elle fournit le service ICT (Information and Communication Technology). Sa taille est le paramètre le plus important pour déterminer les besoins en énergie d'un data center.

Une ferme de serveurs se compose de racks, ces derniers contenant des dizaines de serveurs [Rahmani, 2018].

#### 3.2. Systèmes d'alimentation électrique

La fiabilité et la disponibilité de l'alimentation électrique sont des enjeux critiques pour les data centers. Pour assurer une alimentation électrique ininterrompue, les data centers sont équipés d'un système d'alimentation électrique complexe et redondant. L'électricité est fournie par un transformateur haute tension qui alimente une unité d'alimentation sans coupure (UPS: *Uninterruptible Power Supply*). Les UPS sont capables de fournir l'électricité au data center pendant une courte période, le temps que les générateurs de secours démarrent en cas de défaillance de l'alimentation principale. Les équipements informatiques sont alimentés par des unités de distribution d'énergie (PDU: *Power Distribution Unit*) qui régulent les tensions en fonction de la demande du rack de serveurs.

Pour faire face à une éventuelle défaillance d'approvisionnement en énergie, chaque data center est équipé d'un système d'alimentation de secours, tel que des générateurs diesel. L'alimentation de secours doit permettre de maintenir les fonctions vitales du data center pendant une durée raisonnable,

généralement 72 heures. Les data centers peuvent également utiliser des systèmes de stockage d'énergie, tels que des batteries, pour assurer une alimentation de secours instantanée sans délai de démarrage.

### 3.3. Les systèmes de refroidissement

Les centres de données génèrent une grande quantité de chaleur qui doit être évacuée afin d'éviter que la température ne devienne critique pour les serveurs. Deux systèmes de circulation de la chaleur sont habituellement utilisés : les unités de traitement de l'air des salles d'ordinateurs (CRAH: (*computer room air handler*)) et les unités de climatisation des salles d'ordinateurs (CRAC: (*computer room air conditioning*)). Les unités CRAC sont équipées de leurs propres condenseurs pour réduire la température de l'air, tandis que les unités CRAH nécessitent un système de refroidissement externe pour être alimentées en eau glacée. Leur tâche principale est de maintenir la température ambiante des serveurs entre 20 et 25 °C, avec une plage admissible allant de 15 à 32 °C [Martinez, 2001].

Les centres de données sont également équipés d'un système de refroidissement, qui est souvent basé sur une installation de refroidissement d'eau. Différentes techniques sont utilisées pour transférer la chaleur entre le data center et le refroidisseur. Les paramètres climatiques tels que la température ambiante ont un impact important sur la consommation d'énergie de l'unité de refroidissement. Certains centres de données, comme celui de Facebook à Lulea (en Suède), utilisent l'air naturellement froid pour alimenter les salles d'ordinateurs, ce qui en fait l'un des plus efficaces et durables à ce jour [Dai, 2014].

Par conséquent, un data center se compose d'éléments très complexes, dont les interconnexions multiples et de natures (physiques) différentes rajoutent à la complexité globale du système, dépassant ainsi la simple fonctionnalité de stockage de données. De ce fait, l'identification et la maîtrise des causes qui pourraient altérer le bon fonctionnement du système, voire engendrer une indisponibilité du data center est une tâche très ardue. Il est vrai qu'il existe des méthodes qui permettent de modéliser ce type de système complexe (la modélisation multi-physique par exemple), mais toute la subtilité reste dans le fait d'être capable de modéliser un système complexe avec des méthodes et modèles qui ne sont pas systématiquement du même ordre de complexité, mais qui permettent de répondre convenablement au besoin tout en étant sobre et efficace.

Ce constat a motivé l'orientation de nos travaux vers des approches permettant d'une part de modéliser le comportement de composants, et d'autre part de représenter les interconnexions entre ces derniers (échange de données bidirectionnel).

## 4. MODÉLISATION COMPORTEMENTALE D'UN DATA CENTER

### 4.1. Méthodologie

Dans cette section, nous exposons notre méthode de conception d'un jumeau numérique basée sur des techniques de modélisation comportementale.

Cette méthodologie proposée repose sur les étapes suivantes (cf. Figure 2).

- création de base de connaissances ;
- création des bibliothèques de composants ;
- construction du système ;
- simulation et analyse des résultats.

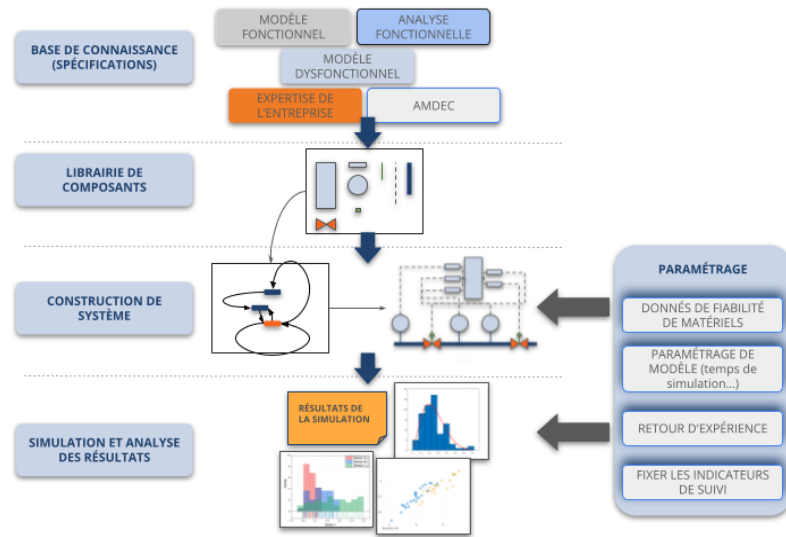


Figure 2. Méthodologie

### 4.2. Création de la base de connaissances

La base de connaissance correspond à l'ensemble des connaissances nécessaires à l'élaboration du JN. Il s'agit en pratique de documentation sur les processus techniques, les analyses fonctionnelles et dysfonctionnelles des différents composants intervenant dans un data center. Autrement dit, cette phase permet de formaliser les objets qui seront simulés par le JN en précisant les règles permettant de relier ces objets entre eux, la structure de décomposition, les relations entre les composants, les différents paramètres permettant de caractériser l'état d'un composant, toutes les transitions permettant les changements d'état, ainsi que les règles de propagation des effets causés par un changement d'état d'un objet du système.

### 4.3. Création de bibliothèque des composants

Cette étape consiste à construire une bibliothèque de modélisation permettant de décrire formellement le comportement fonctionnel et dysfonctionnel de chaque composant du système cible. Chaque composant est caractérisé par des flux d'entrées et de sorties et des automates stochastiques

régissant l'évolution de son état interne. La librairie est structurée autour de classes abstraites décrivant certains comportements génériques (objets électriques, signaux, bloqueur de flux, etc) facilitant ainsi la réutilisation de propriétés communes et par là même la description de composants plus complexes. Par exemple, l'objet "pompe" hérite du comportement fonctionnel des classes abstraites "objet électrique" et "objet hydraulique".

#### 4.4. Construction du système

L'objectif de cette étape est de connecter les composants du système de manière à construire le modèle du système à étudier. Dans le cas d'un data center, les composants décrits dans la Section 3. sont instanciés et reliés entre eux selon les caractéristiques du data center étudié. Cette étape permet d'obtenir *in fine* un modèle simulable du système cible dont le comportement et l'architecture est fidèle à la réalité physique de ce dernier.

#### 4.5. Simulation et analyse

##### 4.5.1. Définir les indicateurs clés à quantifier

Cette étape vise à définir les indicateurs clés que le JN doit prévoir afin de répondre aux problématiques opérationnelles considérées.

Dans le cadre de l'étude d'un data center, nous avons choisi de nous intéresser aux indicateurs de type FMD (Fiabilité, Maintenabilité et Disponibilité) tel que le taux de disponibilité. Ce dernier indique dans quelle mesure un équipement est capable d'exécuter une fonction requise dans des conditions de temps précises et selon un état déterminé. Dans le cas de notre étude, la disponibilité du data center dépend fortement du bon fonctionnement de la chambre des serveurs, ce temps de bon fonctionnement est lié à son tour aux différents paramètres fiabilistes de chaque composant. Ce taux de disponibilité dépend également de plusieurs aspects :

- nombre de pannes (fiabilité) ;
- vitesse de réparation (maintenabilité) ;
- procédures définies (organisation de la maintenance) ;
- qualité des moyens (logistique).

##### 4.5.2. Simulation

La simulation permet de jouer les scénarios de fonctionnement du système dans son ensemble en remontant les résultats des indicateurs établis auparavant. Souvent à ce stade de modélisation, on lance des tests pour vérifier que le fonctionnement observé est conforme aux attentes dans des situations d'utilisation réelles.

Le système est simulé par la technique de Monte-Carlo. Plusieurs séquences de fonctionnement du système sont jouées sur un horizon temporel donné afin d'estimer statistiquement

l'évolution future des indicateurs clés considérés.

La simulation permet également d'explorer les séquences d'événements permettant d'aboutir à un événement donné (par exemple un événement redouté comme la perte de la chambre des serveurs). Cette exploration de séquences permet d'obtenir une hiérarchisation des scénarios menant aux différents événements cibles selon leur probabilité d'occurrence.

#### 4.6. Le framework PyCATSHOO

##### 4.6.1. Motivations techniques

Plusieurs langages de modélisation dynamique sont employés pour la réalisation d'analyses MBSA avancées, notamment AltaRica 2.0 et 3.0 [Prosvirnova et al., 2022], Modelica [Frédéric et al., 2022], AADL [Brunel et al., 2017] et Figaro [Bouissou et al., 1991]. Toutefois, le framework PyCATSHOO a été choisi dans le cadre des travaux présentés du fait de plusieurs avantages clés. PyCATSHOO est une bibliothèque dédiée à la simulation et l'analyse de systèmes hybrides, régis par des comportements déterministes et aléatoires [Chraïbi, 2018]. Elle permet de modéliser avec précision les interactions entre les composants d'un système et son environnement, en tenant compte de leurs caractéristiques déterministes (e.g. propagation de flux dans le système) et aléatoires (e.g. défaillances des composants). L'intégration avec l'écosystème Python facilite la manipulation et la visualisation des données ainsi que l'application de techniques de machine learning. En outre, PyCATSHOO est un outil en développement actif et mis à disposition gratuitement par EDF Lab.

##### 4.6.2. Principaux concepts

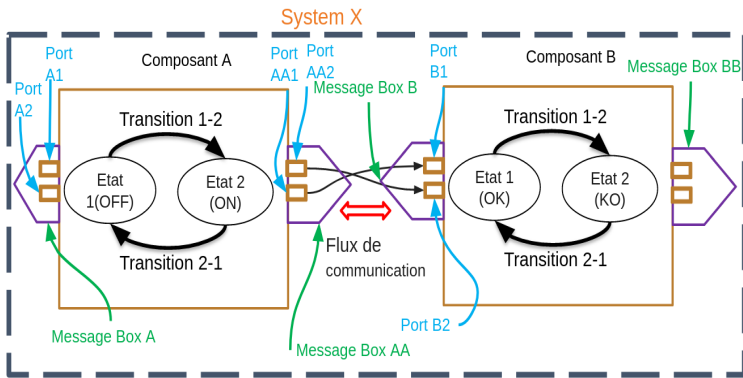
La modélisation d'un système avec PyCATSHOO reposent sur les concepts de variables, références, messages, box, automate, états et transitions (cf. Tableau 1).

**Table 1. Principaux concepts du framework PyCATSHOO**

Concepts	Définition
<b>Variables</b>	Les variables caractérisent l'état intrinsèque d'un composant (niveau d'eau, température, tension, temps, etc.)
<b>Références</b>	Les références sont des réceptacles pour les informations externes provenant de l'extérieur du composant. Il peut s'agir par exemple des valeurs de débit entrant mesurées par un capteur.
<b>Messages Box</b>	Un composant peut présenter plusieurs Messages box. Un message box contient la liste des variables d'un composant dont les

	valeurs sont lisibles à l'extérieur du composant. Elle fournit également des emplacements pour les valeurs entrantes qui alimentent les références du composant.
<b>Automates</b>	Un automate est constitué d'un ensemble d'états. Les automates sont le moyen utilisé par PyCATSHOO pour modéliser le comportement interne dynamique aléatoire d'un composant.

La Figure 3 illustre les différents concepts du framework de modélisation PyCATSHOO. Le système représenté est composé de deux composants, A et B. Les flux sortants de A sont des entrées pour B. Ces deux composants sont connectés *via* leurs message box. Chaque message box peut contenir plusieurs ports, chaque port portant des informations sous forme de variables. Le comportement des composants peut être décrit en utilisant des états, par exemple "marche" ou "arrêt" ou "fonctionnement" ou "panne". Les transitions représentent les événements de franchissement d'un état à un autre, qui peuvent être déclenchés par des conditions.



**Figure 3. Modélisation d'un système à deux composants sous PYCATSHOO**

## 5. APPLICATION

### 5.1. Cas d'étude

La méthodologie de modélisation décrite dans cet article est illustrée sur la construction d'un modèle comportemental relatif au JN d'une architecture de data center telle que présentée en section 3. L'objectif principal de cette étude de cas est de prédire les performances du data center, ainsi que les défaillances potentielles sur un horizon temporel donné. Nous avons décidé de focaliser notre analyse sur la chambre des serveurs, étant donné sa nature critique au sein du data center. Le JN développé

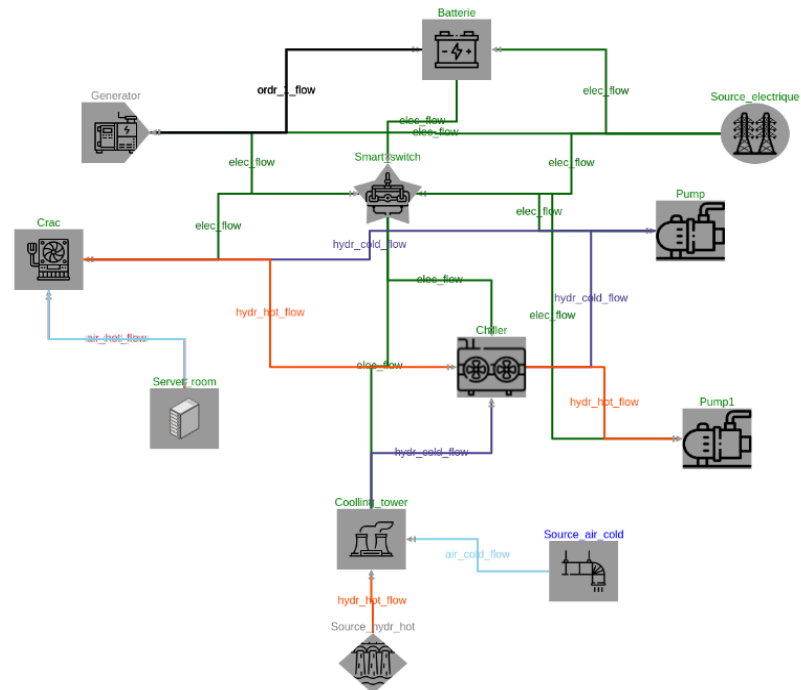
a ainsi pour tâche d'estimer la disponibilité de la chambre des serveurs en considérant l'événement redouté suivant : interruption du système de refroidissement sur une durée de 5 heures.

### 5.2. Hypothèses de modélisation

La Figure 4 illustre le modèle développé, en précisant pour chaque couleur un type de flux circulant dans le modèle. Par exemple, la couleur verte représente un flux de dépendance électrique. Les hypothèses suivantes sont considérées :

- Un composant est disponible s'il est alimenté par tous ses flux en entrée.
- Une défaillance intrinsèque d'un composant stoppe la propagation de ses flux en sortie.
- Les modes de défaillance et de réparation considérés sont régis par des lois d'occurrence exponentielle.

Les paramètres de fiabilité (i.e. taux de défaillance et de réparation) utilisés dans cet étude de cas proviennent de l'article [OREDA, 2002] qui intègre une base de données de fiabilité recensant des informations sur des matériels utilisés dans les data centers issus de retours d'expériences internes ou de données publiées à l'international.

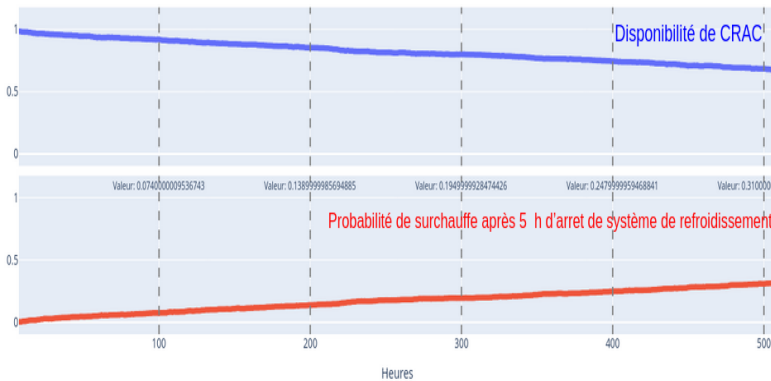


**Figure 4. Modélisation fonctionnelle des flux entre composants dans le jumeau numérique**



### 5.3. Résultats

Les résultats de notre étude de cas ont démontré l'efficacité du modèle de jumeau numérique élaboré dans la prédiction de la disponibilité de la chambre des serveurs en cas d'interruption du système de refroidissement pour une durée de 5 heures, sur un horizon de temps de 500 heures. La Figure 5 présente la courbe en bleu qui illustre la prédiction de la disponibilité du CRAC, qui est en dépendance fonctionnelle directe avec la chambre des serveurs. La courbe en rouge, quant à elle, expose la probabilité de surchauffe de la chambre des serveurs après l'arrêt du système de refroidissement en raison d'un dysfonctionnement. Il convient de noter que durant les cinq premières heures, la probabilité de surchauffe reste nulle, ce qui est conforme aux attentes. Par la suite, la probabilité de surchauffe augmente légèrement.



**Figure 5. Prédiction de la disponibilité de CRAC, salle du serveur sur un horizon 500 heures.**

Grâce au modèle développé, nous avons pu explorer les différentes séquences d'événements qui pourraient conduire à une surchauffe de la chambre des serveurs en cas d'arrêt du système de refroidissement. Cette exploration nous a permis de hiérarchiser les différents scénarios en fonction de leur probabilité d'occurrence, en effectuant plusieurs simulations.

Le Tableau 2 montre les séquences obtenues. La première colonne donne le numéro de la séquence. La 2ème colonne donne une estimation de la probabilité d'occurrence du scénario. Les 3ème et 4ème colonne présentent le temps moyen d'occurrence des événements de la séquence ainsi qu'une description de ces événements.

**Table 2. Séquences obtenues avec PyCATSHOO**

# Séq.	Prob. occ.	Temps moyen occ.	Description de l'événement
1	0.007	355.711	Défaillance de la pompe à l'instant 355.71 h
		360.711	Déclenchement de l'état chaud après 5 heures d'arrêt refroidissement

2	0.004	126.351	Défaillance du chiller à l'instant 126.35 h
		131.351	Déclenchement de l'état chaud après 5 heures d'arrêt refroidissement
3	0.003	295.529	Perte de la source principale et démarrage des batteries
		305.529	Démarrage des groupes électrogènes et arrêt des batteries
		312.315	Réparation de la source principale et arrêt des groupes électrogènes
		433.184	Pannes de source électrique et alimentation est assurée par les batteries
		436,729	Réparation de la source principale et arrêt des batteries
		723.234	Défaillance du cooling tower
		728.234	Déclenchement de l'état chaud après 5 heures d'arrêt refroidissement

Ces séquences permettent aux pilotes du système d'identifier les scénarios critiques lorsque le système est dans un état donné. Ces résultats peuvent ainsi être exploités afin de mettre en place des plans d'actions ou des protocoles visant à anticiper et à protéger le système face à l'occurrence d'événements redoutés.

### 6. CONCLUSION ET PERSPECTIVES

Les travaux présentés dans cet article visent à répondre à la problématique scientifique de l'intégration cohérente et efficace des approches MBSA et MBSE pour la conception et l'évaluation des jumeaux numériques orientés vers la sûreté de fonctionnement. À cette fin, nous avons proposé une méthode de construction d'un modèle comportemental de data center basée sur une approche MBSA/MBSE avancée, en utilisant le framework PyCATSHOO.

Les résultats obtenus ont permis de représenter et de simuler les fonctionnements et dysfonctionnements des composants d'un data center typique. Le jumeau numérique développé a tendance de mieux comprendre les points de faiblesse du système, d'identifier les scénarios de défaillance potentiels et de prévoir les indicateurs de performance sur un horizon de temps donné.

La prochaine phase du projet consiste à connecter le modèle développé avec le système physique. Cela permettra de synchroniser les paramètres du modèle en utilisant les données de capteurs des composants et d'initier automatiquement les simulations pour détecter les futures défaillances potentielles. De

cette manière, le modèle sera capable d'orienter automatiquement les simulations vers des scénarios plus pertinents tout en ignorant les événements qui ne se produisent pas en temps réel.

## 7. RÉFÉRENCES

- Bouissou, M., Bouhadana, H., Bannelier, M., & Villatte, N. (1991). Knowledge Modelling and Reliability Processing : Presentation of the Figaro Language and Associated Tools. *IFAC Proceedings Volumes*, 24, 69-75. [https://doi.org/10.1016/S1474-6670\(17\)51368-3](https://doi.org/10.1016/S1474-6670(17)51368-3)
- Brunel, J., Feiler, P., Hugues, J., Lewis, B., Prosvirnova, T., Seguin, C., & Wrage, L. (2017). Performing Safety Analyses with AADL and AltaRica. 10437, 67. [https://doi.org/10.1007/978-3-319-64119-5\\_5](https://doi.org/10.1007/978-3-319-64119-5_5)
- Chraïbi, H. (2018). Getting started with PyCATSHOO V1.2.2.8 Document version V1.
- Collart-Dutilleul, S. (2009). Réseaux de Petri P-temporels : Modélisation et validation d'exigences temporelles.
- Dai, J., Ohadi, M. M., Das, D., & Pecht, M. G. (2014). Optimum Cooling of Data Centers. Springer New York. <https://doi.org/10.1007/978-1-4614-5602-5>
- Demachy, R., & Guilmeau, S. (2022, juin 1). Structural consistency of MBSE and MBSA models using Consistency Links. 11th European Congress Embedded Real Time System ( ERTS 2022 ). <https://hal.science/hal-03697170>
- Donat, R., Legendre, A., & Perez, L. (s. d.). Démarche d'analyse MBSA dynamique pour la réalisation d'études FMD sur une ligne de transport ferroviaire complète.
- Douchet, F. (2015). Data center energy optimization using liquids to cool computer racks [Phdthesis, Université de Bretagne Sud]. <https://theses.hal.science/tel-01325363>
- Estefan, J. A. (2008). Survey of Model-Based Systems Engineering (MBSE) Methodologies.
- Frédéric, M., Michel, B., & Xavier, D. B. (2022). Model-Based Safety Assessment : Comment renforcer la confiance dans les modèles ?
- Lebreton, P. (2016). Datacenter-avec-CARL-Source.pdf. <https://www.carl-software.fr/wp-content/uploads/2016/10/20151218-143951elecCityGroup-assure-la-maintenanc-e-de-ses-Datacenter-avec-CARL-Source.pdf>
- Li, Y., Gong, Q., & Su, D. (2014). Model-based System Safety Assessment of Aircraft Power Plant. *Procedia Engineering*, 80, 85-92. <https://doi.org/10.1016/j.proeng.2014.09.063>
- Martinez, C. (s. d.). ASHRAE HVAC 2001 Fundamentals Handbook.pdf. Consulté 14 avril 2023, à l'adresse [https://www.academia.edu/26602849/ASHRAE\\_HVAC\\_2001\\_Fundamentals\\_Handbook\\_pdf](https://www.academia.edu/26602849/ASHRAE_HVAC_2001_Fundamentals_Handbook_pdf)
- Pearl, T. (2003). Tier Certification Overview. <https://uptimeinstitute.com/tier-certification>
- Prosvirnova, T., Seguin, C., Frazza, C., Batteux, M., de Bossoreille, X., Deschamps, F., Gauthier, J., & Saez, E. (2022). Strategies for Modelling Failure Propagation in Dynamic Systems with AltaRica. In C. Seguin, M. Zeller, & T. Prosvirnova (Éds.), *Model-Based Safety and Assessment* (Vol. 13525, p. 101-115). Springer International Publishing. [https://doi.org/10.1007/978-3-031-15842-1\\_8](https://doi.org/10.1007/978-3-031-15842-1_8)
- Rahmani, R., Moser, I., & Seyedmahmoudian, M. (2018). A Complete Model for Modular Simulation of Data Centre Power Load (arXiv:1804.00703). arXiv. <https://doi.org/10.48550/arXiv.1804.00703>
- Vidalie, J., Batteux, M., Choley, J.-Y., Mhenni, F., & Kendel, M.-S. (2020). Typology of the differences Between Model-Based System Engineering (MBSE) and Safety Assessment (MBSA) models : Analysis of a Reference System.