



HAL
open science

Schéma de co-simulation pour l'équilibrage des échanges d'énergie dans un micro-réseau

Jean-Baptiste Wiart, Vincent Chevrier, Denis Netter

► **To cite this version:**

Jean-Baptiste Wiart, Vincent Chevrier, Denis Netter. Schéma de co-simulation pour l'équilibrage des échanges d'énergie dans un micro-réseau. JFMS 2022 - Journée Francophone de la Modélisation et de la Simulation, Mar 2022, Cargèse, France. hal-03699965

HAL Id: hal-03699965

<https://hal.science/hal-03699965>

Submitted on 20 Jun 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Schéma de co-simulation pour l'équilibrage des échanges d'énergie dans un micro-réseau

co-simulation scheme for balancing energy exchanges in a microgrid

Jean-Baptiste WIART^{1,2}

Vincent CHEVRIER¹

Denis NETTER²

¹ Université de Lorraine, CNRS, LORIA, F-54000 Nancy, France

² Université de Lorraine, GREEN, F-54000 Nancy, France

prénom.nom@loria.fr¹

prénom.nom@univ-lorraine.fr²

Résumé :

Le dimensionnement et la gestion de systèmes physiques, et particulièrement de microgrilles, nécessitent l'utilisation de modèles et de simulateurs relatifs aux domaines de télécommunication, énergétique ou décisionnel; c'est un contexte privilégié pour la co-simulation. Cependant l'essence même des interactions entre modèles physiques, conditionnée par un système décisionnel, peut entraîner des modifications dans le multi-modèle implémenté.

Nous montrons ici qu'un traitement ad-hoc limite l'approche itérative et modulaire. Nous proposons une (multi-)modélisation du problème qui permet de manière aisée de réaliser des co-simulations de microgrid testant différentes configurations de réseaux et de système de décision.

Après avoir mis en évidence le problème à partir d'exemples représentatifs, nous présentons notre contribution puis illustrons son intérêt en reprenant les exemples dans le cadre de l'intergiciel MECSYCO.

Mots-clés :

Multi-modélisation, co-simulation, échanges d'énergie

Abstract:

The design and management of physical systems, and particularly of microgrids, require models and simulators related to the telecommunication, energy or decision domains; it is a privileged context for co-simulation. However, the structure of the interactions between physical models is conditioned by the decision system and leads to modifications in the implemented multi-model.

We show that an ad-hoc treatment limits iterative and modular approaches. We propose a (multi-)modeling of the problem that allows to easily implement co-simulations of microgrids testing different network and decision system configurations.

After highlighting the problem from examples, we present our contribution and illustrate its interest by taking the examples in the middleware framework MECSYCO.

Keywords:

Multi-modeling, co-simulation, energy exchanges

1 Introduction

L'intégration et la gestion de nouvelles sources d'énergie renouvelable est un enjeu majeur de la transition énergétique. L'intermittence des sources rend la prédiction de la consommation et de la production indispensable pour équilibrer les échanges d'énergie entre producteurs et consommateurs [1]. Les microgrids permettent de gérer localement cet équilibre [3], en particulier à l'aide de systèmes de gestion, étudiés sous différents aspects (coûts, émissions de gaz à effet de serre, pics de charge) [9]. En raison des questions interdisciplinaires que ces systèmes soulèvent, la multi-modélisation et co-simulation constituent une approche d'analyse efficace par rapport à la construction de banc d'essai [7].

Cependant, dans le cas des microgrilles, systèmes cyber-physiques (CPS), la structure des échanges d'énergie entre producteurs et consommateurs est dynamique. La représentation ad-hoc de cette structure dans une co-simulation est rapidement limitée lors du passage à l'échelle [4].

Notre proposition dans cet article est une multi-modélisation générique de cette structure d'échange d'énergie. Nous détaillons d'abord le contexte et la problématique en l'illustrant par des exemples. Nous détaillons ensuite notre proposition et illustrons son utilisation avec l'intergiciel MECSYCO avant de conclure.

2 M&S de micro-réseaux

2.1 Généralités

Les CPS nécessitent une approche de co-simulation pour leur étude, notamment dans le domaine de la transition énergétique où trois domaines sont souvent impliqués [5].

MECSYCO¹ est un intergiciel de co-simulation s'appuyant sur une stratégie de wrapping DEVS pour l'intégration logicielle et formelle et sur un méta-modèle AA4MM (Agent and Artifact for Multi-Modeling) [2] qui envisage la simulation comme des interactions entre agents. L'approche développée dans MECSYCO se veut générique, dans le sens où plusieurs problèmes liés à la co-simulation sont résolus : synchronisation (simulateur abstrait DEVS), composition (fermeture par couplage en DEVS), modularité (équivalence de modèle au sens de SES [11] et notion d'encapsulation), etc.

Toutefois, l'approche se focalise en premier lieu sur des problèmes de co-simulation mais ne prend pas en compte les aspects métiers du système [10] à simuler. En outre, les approches de co-simulation dédiées font encore l'objet de recherches et sont souvent liées à des projets de recherche spécifiques ainsi qu'à des développeurs spécifiques [6]. En conséquence, la modularité est limitée par les interfaces des modèles et par le manque de représentation de certains aspects du système comme les échanges d'énergie.

2.2 co-simulation avec MECSYCO

Cette partie reprend les éléments essentiels d'une démarche de construction de co-simulation définis dans [8] et permettent de comprendre les principes des schémas utilisés dans l'article.

Une étape préliminaire est de wrapper les modèles/simulateurs formellement et logiciellement dans MECSYCO pour qu'ils puissent être

considérés comme des modèles atomiques au sens de DEVS. Cela permet d'identifier l'interface de chacun des modèles utilisables sous la forme de ses entrées et sorties.

Construire un schéma de co-simulation revient à un couplage de certains de ces modèles réalisé en mettant en relation les sorties des uns avec les entrées des autres.

La co-simulation de ce schéma passe par son instanciation en MECSYCO grâce à un DSL². L'exécution (et la synchronisation) des simulateurs revient alors à exécuter un système DEVS et est assurée dans l'intergiciel par son algorithme de simulation.

Ainsi sur le schéma de co-simulation de la figure 1, nous avons trois modèles wrappés, représentant la météo (ici une base de données), un modèle thermique d'une maison (un export FMU à partir de Modelica) et un contrôleur de climatiseur (en Matlab/Simulink). La température du bâtiment évolue en fonction de la température extérieure, reçue du modèle météo. La régulation est fonction d'une température de consigne donnée à un contrôleur. Le contrôleur renvoie la puissance de fonctionnement du climatiseur. Chaque composant est équivalent à un modèle DEVS atomique qui émet et reçoit des événements. Le modèle météo est une source d'événements qui sont transmis au modèle de maison. Ce modèle réagit en mettant à jour sa température intérieure, transmise régulièrement au contrôleur, lequel réagit en indiquant en retour une valeur de consigne pour le climatiseur.

2.3 Le problème d'équilibrage des échanges d'énergies dans les micro-réseaux

Pour illustrer les difficultés liées au problème d'équilibrage des échanges d'énergies, nous prenons pour exemple une micro-structure. Nous souhaitons dimensionner un micro-réseau

1. disponible sur mecsycos.com

2. Le lecteur intéressé peut consulter cet exemple sur <https://gitlab.inria.fr/Simbiot/mecsycos/mecsycoscholar>.

électrique à l'échelle d'une habitation pour la rendre autonome en énergie. Nous reprenons comme point de départ le multi-modèle défini en figure 1, donnant par simulation, la puissance nécessaire pour climatiser l'habitation.



Figure 1 – Schéma de co-simulation de régulation thermique.

Nous souhaitons ajouter des sources d'énergie renouvelable (photovoltaïque, éolien) puis des moyens de stockage (batterie, hydrogène) pour au final obtenir un dimensionnement du système (notamment la taille des stockages) et une estimation de la quantité d'énergie devant être achetée sur le réseau électrique (et donc savoir si la maison est autonome en énergie ou pas).

Cette quantité d'énergie est dépendante de la météo qui influence la température de la maison et les productions éolienne et photovoltaïque. Comme cette quantité d'énergie varie durant la simulation, il faut modifier les flux d'énergie entre charges, sources et stockages pour toujours assurer l'équilibre global du système. Ainsi, en considérant la maison, une source d'énergie renouvelable, un stockage et le réseau électrique, plusieurs schémas, illustrés sur la figure 2, sont à considérer selon la situation :

- a. la source renouvelable produit plus que le besoin de la maison et peut donc alimenter le stockage sans faire intervenir le réseau électrique ;
- b. la source renouvelable produit moins que le besoin de la maison et le stockage (suffisant) doit donc aussi alimenter la maison sans faire intervenir le réseau électrique ;
- c. la source renouvelable produit moins que le besoin de la maison et le stockage (insuffisant) doit alimenter la maison ainsi que le réseau électrique.

Ces schémas (n'indiquant que des flux entre modèles) sont insuffisants pour rendre compte du problème d'équilibre des échanges. En effet, il manque les valeurs de ces échanges. Une modification dans l'une de ces valeurs (par exemple la quantité d'énergie consommée par la maison) oblige à déterminer une nouvelle répartition sur l'ensemble du multi-modèle. Face à ce problème, des personnes ayant une bonne connaissance de l'outil l'ont résolu en utilisant les possibilités de programmation de l'intergiciel de co-simulation mais en mélangeant flux d'information (besoin en énergie de la maison) et flux d'énergie (production de la source), pouvant mener à des aberrations comme le non-respect des lois physiques³.

Le principe sous-jacent utilisé pour résoudre le problème, illustré sous la figure 3, est d'ajouter au multi-modèle des flux d'informations et des calculs, sous la forme d'aggrégateurs et comparateurs (outils propres à MECASYCO.) pour déterminer les flux et leurs valeurs garantissant un équilibre (par exemple savoir que l'on passe du cas 1 au cas 2).

Cette façon de faire ad-hoc a permis des co-simulations fonctionnelles mais uniquement sur des sous-ensembles de cas limités et demande à revoir le schéma de co-simulation pour chaque cas non encore traité. Des experts en énergie, capables de construire des schémas de co-simulation avec des grandeurs physiques (comme pour la figure 1), seront incapables de faire cela car demandant beaucoup de connaissances de l'outil. De plus, le passage à l'échelle, avec un nombre de composants conséquent, est rendu difficile puisque plus il y a de composants, plus il faut ajouter d'éléments de comparaison et d'aggrégation.

En résumé, cette approche ad-hoc fait appel à des connaissances des outils mis à disposition par MECASYCO. (et donc incompréhensible si l'on n'est pas expert, la figure 3 en est un bon

3. Le lecteur intéressé trouvera ces exemples sur <https://gitlab.inria.fr/Simbiot/mecsyco/mecsyco-scholar>.

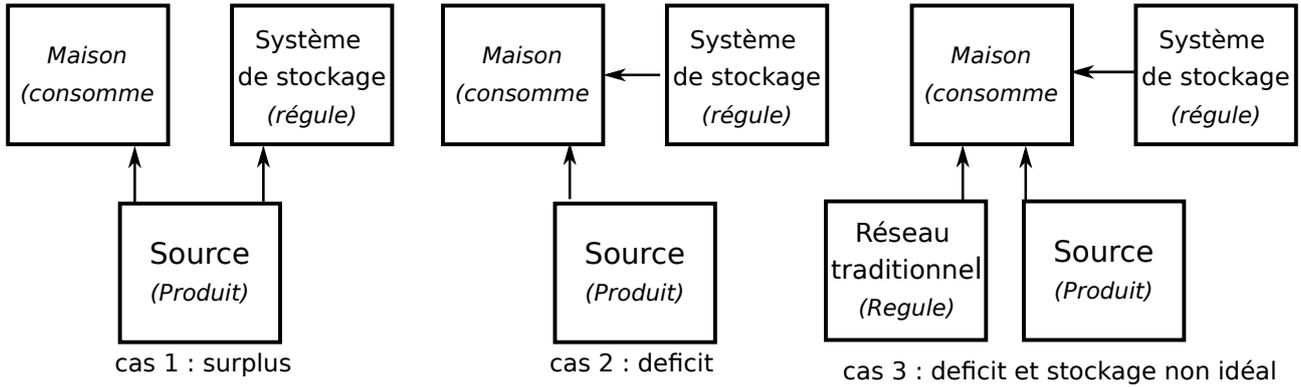


Figure 2 – Echange d'énergie du micro-réseau selon différentes situations.

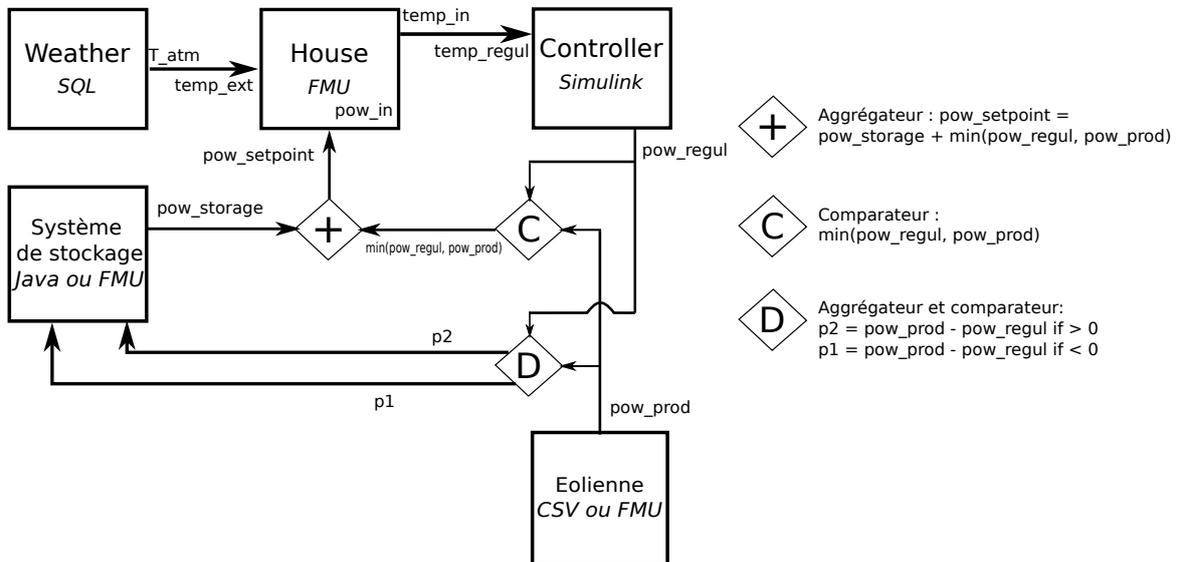


Figure 3 – Schéma de co-simulation ad-hoc d'une maison autonome.

exemple); code la stratégie avec ces mêmes outils et par conséquent rend difficile toute vérification ou modification); et enfin n'est utilisable que si la structure du schéma n'évolue pas, auquel cas il faut tout refaire.

Il manque donc un schéma de co-simulation dédié à ce problème (fréquent dans le domaine énergétique), capable de passer à l'échelle, permettant de représenter explicitement la stratégie de gestion des flux afin de fournir au niveau de l'outil des primitives qui relèvent du domaine et dont la sémantique est appréhendable par des énergéticiens

3 Définition d'un schéma de co-simulation générique

Notre proposition consiste en deux principes :

- réifier la stratégie de gestion comme un modèle de la co-simulation (figure 4). Son rôle vis-à-vis du problème est de mettre en œuvre la stratégie de gestion en décidant les productions/consommations de chacun des autres composants du système.
- considérer les autres modèles physiques sous leur angle informationnel (que nous désignerons sous le terme de *smart-composants*).

Ces modèles et la stratégie de gestion n'échangeront que des événements cor-

respondant à des informations (et non des grandeurs physiques).

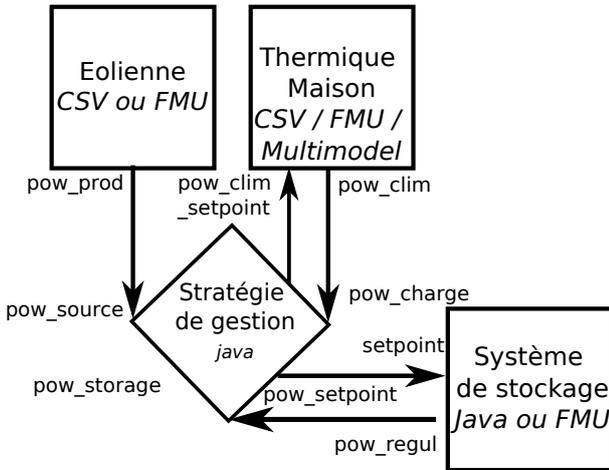


Figure 4 – Réification de la stratégie de gestion d'énergie

3.1 Principes

Le système est vu comme un ensemble de composants (modèles de source, de système de stockage, ou de charge) informationnels : nous nous plaçons au niveau contractuel du réseau énergétique [1]. Ainsi nous supposons que les problématiques physiques électriques des composants sont respectées (équilibre fréquence, tension) et nous raisonnons en puissance.

Chaque composant possède les capteurs nécessaires pour connaître la quantité d'énergie qu'il produit et/ou consomme.

Ces hypothèses nous permettent de définir une interface pour les éléments du système.

Composants du système. Les smart-composants connaissent leur puissance consommée ou produite. Chacun est défini par son rôle (production, consommation, stockage) et possède un ensemble de comportements possibles. Les valeurs de puissances, valeurs à dynamique continue, sont, selon le modèle, discrétisées ou varient de manière discrète.

Un premier comportement pour un composant est d'informer sur son état, c'est-à-dire indiquer la puissance consommée ou produite par

celui-ci à un instant donné. Cela se traduit par le fait que tout changement d'état vis à vis de la puissance correspond à la génération d'un événement d'information (à destination de la stratégie de gestion).

Le second comportement est de pouvoir adapter sa consommation/production. Cela se traduit par la connaissance des valeurs de consommation/production acceptables et la possibilité de s'adapter à des consignes. Ce comportement est mis en œuvre lors du changement d'état par l'émission d'événement informant, outre le nouvel état de puissance, des niveaux de puissances possibles, et par la réception d'événement de consigne (pour par exemple imposer une production à une source). Le niveau minimal de ce second aspect est la possibilité de déconnexion : un smart-composant n'ayant aucun degré de liberté possède tout de même la possibilité de se déconnecter du réseau si la stratégie le lui demande (cela correspond à une réalité physique).

Un smart-composant propose donc l'interface suivante :

- une entrée qui correspond à une consigne de puissance : $I_c \in \mathbb{R} \cup null$, si la valeur est absente (null), l'adaptation n'est pas nécessaire et la valeur de puissance du composant restera donc à la valeur actuelle, sinon la valeur est soit l'une parmi les valeurs acceptables, soit 0 qui signifie la déconnexion du composant de la grille.
- une sortie qui correspond à la puissance consommée ou produite suite à un changement d'état : $O_c \in \mathbb{R}$
- une sortie qui correspond aux valeurs de puissances consommées ou produites acceptables : $O_{c_i} \subset \mathbb{R}$

Une valeur positive signifie que le composant est une source, et sinon une charge. Dans le cas d'un stockage, la valeur peut prendre soit positive (le stockage produit), soit négative (il consomme).

Opérateur de couplage décisionnel. Le deuxième élément de notre proposition est le modèle représentant la stratégie de gestion. Son rôle est de gérer les flux entre les composants du système pour maintenir l'équilibre entre offre et demande. Ce modèle maintient une représentation de l'état du réseau sous la forme, pour chaque composant, des puissances (produites et consommées), ainsi que des consignes acceptables. La réaction à une modification de cet état est alors d'établir, selon la stratégie, pour chaque composant sa consigne de puissance (parmi celles acceptables) avec comme contrainte que la somme des flux soit nulle (condition de faisabilité physique).

L'opérateur est couplé à n smart-composants. Nous pouvons en déduire l'interface suivante :

- entrées : Ensemble des informations de puissance : $I_p \in \mathbb{R}^n$
- entrées : Ensemble des consignes de puissances acceptables : $I_{ca} \in \mathbb{R}^n$
- sorties : Ensemble des consignes de puissance envoyées aux smart composants, à nombre variable : $O_c \in \mathbb{R}^p$
- paramètre : la stratégie de gestion (sous la forme d'une fonction : $I_p \times I_{ca} \rightarrow O_c$)

3.2 Déroulement de la co-simulation

Tous nos composants étant des modèles atomiques au sens de DEVS et s'échangeant des événements, leur simulation consiste en l'exécution d'un algorithme de simulation DEVS. Néanmoins, trois remarques s'imposent.

1) L'initialisation de l'opérateur de couplage doit être effectuée, il faut donc un comportement d'initialisation. Nous faisons l'hypothèse que la configuration initiale est à l'équilibre énergétique. Le comportement initial de tous les smart-composants est de générer un événement correspondant à leur puissance initiale. Cela permet de construire la représentation du réseau.

2) L'opérateur de couplage décisionnel étant

un modèle, il calcule les flux d'équilibrage entre deux pas de simulation, paramétré par le modélisateur. Il représente le temps de procéder à la recherche de solution au problème d'équilibrage et va se répercuter dans l'ensemble de la co-simulation.

3) L'absence de solution au problème d'équilibre se traduit par la déconnexion de tous les composants de la grille. Cela permet tout de même d'avoir une solution physiquement acceptable.

3.3 Synthèse

Notre proposition consiste donc en une vision basée port pour l'équilibrage des flux énergétiques dans un système comprenant sources, charges, et systèmes de stockage. Il propose des composants conçus pour ce problème qui permettent de manière générique de construire des schémas de co-simulation dédiés à l'équilibrage.

Ces composants ont été conçus pour permettre des co-simulation modulaires (en autorisant le changement de stratégie ou de composant) et incrémentales (ajouter un composant laisse la logique inchangée). Enfin comme ils repectent le formalisme DEVS dans leur interface, leur intégration dans un écosystème de composants DEVS est immédiate.

4 Illustration de la proposition : le problème de la maison autonome

Nous avons mis en œuvre ces notions dans MECSYCO en implantant la stratégie sous la forme d'un modèle JAVA et en développant quelques smart-composants.

Nous illustrons ici l'utilisation de notre proposition en la comparant avec une approche ad-hoc. L'objectif de la co-simulation est de trouver les conditions nécessaires pour rendre notre maison autonome en énergie. La co-simulation utilisée est celle présentée figure 4⁴.

4. Les codes de ces co-simulations sont dis-

Dans ce multi-modèle, le système de stockage et la maison peuvent recevoir des consignes. Le premier permet de réguler les échanges d'énergie entre les consommateurs et les producteurs. Il est idéal en terme de volume stocké et de dynamique de stockage. La source ne fait qu'imposer sa production. La maison reçoit la consigne de puissance utilisée pour ses besoins (puissance du climatiseur).

Ce premier exemple est restreint ; la source ne peut recevoir des consignes et la puissance est toujours disponible. Par conséquent l'implantation de la stratégie (une fonction Java) est simple, la maison reçoit la puissance demandée en amont. Ce que nous voulons montrer ici est la faisabilité de notre approche (certes restreinte) et son intérêt en terme de facilité pour un thématicien. L'approche ad-hoc et la notre ont des résultats équivalents. En revanche, si l'on compare les descriptions données en figure 5 correspondant à l'approche ad-hoc et l'approche DSL avec notre opérateur, nous notons les différences suivantes :

- Décrire le même système nécessite moins de code DSL et les composants ad-hoc (par exemple celui défini par le mot-clé **agregation**) sont remplacés par un composant modulaire identifié (appelé ici *Operator*).
- L'algorithme de résolution de l'équilibre est identifié et plus compréhensible qu'avant (dans notre cas, il prend la forme d'un code JAVA).
- L'opérateur de couplage étant un modèle, il est observable (on peut coupler ses ports de sorties à des outils de visualisation) contrairement aux composants ad-hoc. L'analyse du système est simplifiée.
- L'interface des composants (smart ou non selon les cas) est homogénéisée. De cette manière, les flux physiques et in-

formationnels ne sont plus mélangés.

Enfin, l'ajout de composants (comme un panneau photovoltaïque) dans le système suit la même logique (figure 6), il suffit de coupler le nouveau composant à l'opérateur lequel se chargera de la répartition des flux d'énergie. Concrètement, cet ajout de composant est décrit dans le DSL MECSYCO par un ajout de déclaration (1 ligne - non visible ici) et un ajout de couplage entre l'opérateur et le nouveau composant (1 ou 2 lignes si le composant peut recevoir ou non une consigne), ici il suffit d'ajouter une ligne (ligne 16).

5 Discussions et perspectives

Dans cet article, nous avons proposé une approche pour la résolution de problématique d'équilibrage énergétique dans un micro-réseau. Cette approche s'appuie sur la réification de la stratégie de gestion sous la forme d'un modèle et le concept de smart-composant.

Cette approche permet d'une part, de séparer les responsabilités : les flux informationnels et physiques ne sont plus mélangés contrairement à une approche ad-hoc. D'autre part, la stratégie de gestion est un modèle, qui laisse la possibilité à l'utilisateur d'aisément changer cette stratégie sans remettre en cause le schéma de co-simulation.

Par ailleurs, même si l'instanciation détaillée est propre à MECSYCO, la définition du concept de smart-composant ainsi que la réification de la stratégie de gestion sous la forme de modèles basés ports rendent possible la réutilisation de l'approche dans d'autres approches de co-simulation.

L'illustration proposée est une preuve de concept. Dans la suite de nos travaux, nous souhaitons appliquer cette approche à des systèmes plus complexes. Notamment intégrer des modèles existants de système de stockage

```

1  internal couplings
2  // Coupling for the house multimodel
3  {weather."temp" -> House."Text"}
4  {House."T1" -> Controller."Templ"}
5
6  // Coupling for the energy flow management system
7  //link between controller and PV (Min) -> Connector1
8  //in order to calculate Min(Pprod, Pcontroller) = Compareur(Min) output (W)
9  {aggregation "operation.CompareurDoubleAggregationOperation", (Controller.
10     "Pow1_out", Eolienne."prodEolien")
11     -> (Connecteur1."input")}
12
13 //link between (controller and Windturbine as DeltaP) -> Storage
14 //Multiplexer in order to calculate DeltaP (Pprod - Pcontroller) = Compareur (
15     DeltaP) (W)
16 //We send the excess of energy to the storage or we send 0
17 {aggregation "mecsyco.core.operation.aggregation.SumAggregationOperation",
18     (Controller."Pow1_out" event operation "operation.MultiplyDoubleOperation"
19     arg=-1., Eolienne."prodEolien")
20     -> (Stockage."productionEnergy" event operation "operation.
21     MaxDoubleOperation" arg=0.)}
22
23 //link between (controller and Windturbine as DeltaP) -> Storage
24 //Multiplexer in order to calculate DeltaP (Pprod - Pcontroller) = Compareur (
25     DeltaP) (W)
26 //We send the deficit of energy to inform the stock that we need power or we send 0
27 {aggregation "operation.aggregation.SumAggregationOperation",
28     (Controller."Pow1_out" event operation "operation.MultiplyDoubleOperation"
29     arg=-1., Eolienne."prodEolien")
30     -> (Stockage."powerReq" event operation "operation.
31     MinDoubleOperation" arg=0.)}
32
33 //Link between compareur and convUP --> House (W)
34 //+=adder: we add all the powers we have
35 {aggregation "operation.aggregation.SumAggregationOperation",
36     (Connecteur1."output",
37     Stockage."energyOutcome" event operation "operation.
38     MultiplyDoubleOperation" arg=-1.)
39     -> (Connecteur2."input")}
40
41 //Power sent to the house
42 {Connecteur2."output" -> House."Pow1" event operation "operation.
43     MultiplyDoubleOperation" arg=-1.}

```

(a) Couplage de la co-simulation ad-hoc.

```

1  internal couplings
2  // Coupling for the load
3  // Coupling between weather and house
4  {weather."temp" -> House."Text"}
5  //Coupling between house and controller
6  {House."T1" -> Controller."Templ"}
7  // Coupling between controller and house
8  {Operator."Charge_power1_out" -> House."Pow1"
9     event operation "operation.MultiplyDoubleOperation" arg=-1.}
10
11 // Coupling for the operator
12 {Controller."Pow1_out" -> Operator."Charge_power1"
13     event operation "operation.MultiplyDoubleOperation" arg=-1.}
14 // Coupling Source and Operator
15 {Eolienne."prodEolien" -> Operator."Source_power1"}
16 // Coupling storage and operator
17 {Operator."Stock_power_out" -> StorageSystem."setPoint"}

```

(b) Couplage de la co-simulation générique.

Figure 5 – Comparaison du couplage entre la cosimulation ad-hoc et celle utilisant l’opérateur proposé.

```

1  internal couplings
2  // Coupling for the load
3  // Coupling between weather and house
4  {weather."temp" -> House."Text"}
5  // Coupling between house and controller
6  {House."T1" -> Controller."Templ"}
7  // Coupling between controller and house
8  {Operator."Charge_power1_out" -> House."Pow1"
9     event operation "operation.MultiplyDoubleOperation" arg
10     =-1.}
11
12 // Coupling for the operator
13 {Controller."Pow1_out" -> Operator."Charge_power1"
14     event operation "operation.MultiplyDoubleOperation" arg
15     =-1.}
16 // Coupling Source(s) and Operator
17 {Eolienne."prodEolien" -> Operator."Source_power1"}
18 {SolarPanel."prodSolar" -> Operator."Source_power2"}
19 // Coupling storage and operator
20 {Operator."Stock_power_out" -> StorageSystem."setPoint"}

```

Figure 6 – Ajout d’un composant via l’opérateur.

à hydrogène, des modèles de consommation d’habitation et réaliser des systèmes à des

échelles différentes. Dans cette co-simulation, le degré de liberté pour résoudre le problème d’équilibrage est connu et unique (le stockage). Les co-simulations suivantes devront apporter une nouvelle dimension aux composants smart : chacun pourra avoir ses degrés de liberté à communiquer à l’opérateur de couplage. Cela demandera à proposer d’autres stratégies.

De plus, nous avons fait l’hypothèse que la dimension physique du réseau (équilibre en fréquence et en tension) est respectée. Il nous faut travailler à lever cette hypothèse ou du moins expliquer comment nous pouvons intégrer sa gestion de manière compatible à notre proposition.

Enfin, MECSYCO permet la composition (au sens de DEVS). Nous souhaitons étendre ce

modèle d'équilibrage pour le rendre composable et ainsi coupler des modèles d'équilibrage entre eux pour aborder des problématiques plus vaste (auto-consommation collective et non uniquement pour une seule maison) avec différents niveaux (maison, quartier, ...). Par ailleurs, nous souhaitons coupler la co-simulation à des méthodes d'optimisation, permettant ainsi d'intégrer la co-simulation dans une démarche plus vaste d'aide à la décision.

Une dernière perspective est de comprendre à quelles conditions un composant physique dans le système cible peut être "smartifié".

Remerciements

Ce travail a bénéficié d'une aide de l'État, gérée par l'Agence Nationale de la Recherche, au titre du projet Investissements d'Avenir Lorraine Université d'Excellence, portant la référence ANR-15-IDEX-04-LUE.

Références

- [1] J. Albuouys-Perrois. *Simulation multi-agent de l'autoconsommation collective de l'énergie à l'échelle du quartier en lien avec l'activité humaine et la consommation énergétique des bâtiments*. PhD thesis, Université de La Rochelle, 2021.
- [2] B. Camus, T. Paris, J. Vaubourg, Y. Presse, C. Bourjot, L. Ciarletta, and V. Chevrier. Co-simulation of cyber-physical systems using a DEVS wrapping strategy in the MECSYCO middleware. *SIMULATION*, 2018.
- [3] S. Chowdhury, S. P. Chowdhury, and P. Crossley. *Microgrids and active distribution networks*. Number 6 in IET renewable energy series. Institution of Engineering and Technology, Stevenage, 2009. OCLC : ocn318673357.
- [4] C. Gomes, C. Thule, D. Broman, P. G. Larsen, and H. Vangheluwe. Co-Simulation : A Survey. *ACM Computing Surveys*, 51(3) :1–33, May 2018.
- [5] E. Hammad, M. Ezeme, and A. Farraj. Implementation and development of an offline co-simulation testbed for studies of power systems cyber security and control verification. *International Journal of Electrical Power & Energy Systems*, 104 :817–826, 2019.
- [6] S. C. Müller, H. Georg, J. J. Nutaro, E. Widl, Y. Deng, P. Palensky, M. U. Awais, M. Chenine, M. Küch, M. Stifter, et al. Interfacing power system and ict simulators : Challenges, state-of-the-art, and case studies. *IEEE Transactions on Smart Grid*, 9(1) :14–24, 2016.
- [7] P. Palensky, A. van der Meer, C. Lopez, A. Joseph, and K. Pan. Applied cosimulation of intelligent power systems : Implementing hybrid simulators for complex power systems. *IEEE Industrial Electronics Magazine*, 11(2) :6–21, Jun 2017.
- [8] T. Paris, J.-B. Wiart, D. Netter, and V. Chevrier. Teaching co-simulation basics through practice. In *2019 SUMMER SIMULATION CONFERENCE*, 2019.
- [9] H. Shayeghi, E. Shahryari, M. Moradza-deh, and P. Siano. A survey on microgrid energy management considering flexible energy sources. *Energies*, 12(11) :2156, 2019.
- [10] M. Vogt, F. Marten, and M. Braun. A survey and statistical analysis of smart grid co-simulations. *Applied energy*, 222 :67–78, 2018.
- [11] B. P. Zeigler, A. Muzy, and E. Kofman. *Theory of modeling and simulation : discrete event & iterative system computational foundations*. Academic press, 2018.