



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE

**UNIVERSITE IBN KHALDOUN - TIARET**

# MEMOIRE

Présenté à :

FACULTÉ MATHÉMATIQUES ET INFORMATIQUE  
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

**MASTER**

Spécialité : Réseaux et télécommunications

Par :

**CHAKLALA Abed  
SADOUK Boubakeur**

Sur le thème

---

## **Optimisation de la consommation d'énergie dans les bâtiments intelligents : Modélisation DEVS et simulation**

---

Soutenu publiquement le 29 / 06 / 2022 à Tiaret devant le jury composé de :

MOSTEFAOUI Sid Ahmed Mokhtar	MCA	Université Tiaret	Président
MAATOUG Abdelfettah	MCB	Université Tiaret	Encadreur
KERMAS Nawel	MCB	Université Tiaret	Examineur
KHAROUBI Sahraoui	MCA	Université Tiaret	Co-encadreur

2021-2022

# *Remerciements*

Tout d'abord, nous voudrions remercier « **ALLAH** » Tout-Puissant et Miséricordieux. Créateur de tout l'univers qui nous a donné la santé, la force, le courage et la volonté de mener à bien cet humble travail de recherche.

Nous voudrions remercier toutes les personnes ayant soutenu et encadré ce travail :

En premier lieu, nous remercions sincèrement et très chaleureusement, nos encadreurs, Mr. **MAATOUG Abdel-Fettah**, Maitre de Conférences classe B à l'université Ibn Khaldoune de Tiaret, et Mr. **KHAROUBI Sahraoui**, Maitre de Conférences classe A à l'université Ibn Khaldoune de Tiaret, pour tout, que ce soit en raison de leur disponibilité ou de leur principale contribution à la direction des travaux de recherche et de leurs idées originales et éclairantes qui ont influencé le contenu de ce mémoire de Master et ouvert la voie à son achèvement.

Nous remercions également Mr. **MAASKRI Mustapha** pour l'aide qu'il nous a apporté et les précieux conseils qui ont abouti à ce travail.

Nous tenons à remercier Mr. **MOSTEFAOUI Sid Ahmed Mokhtar**, Maitre de Conférences classe A à l'université Ibn Khaldoune de Tiaret, pour avoir accepté la présidence du jury,

Nous tenons également à remercier Mme. **KERMAS Nawel**, Maitre de Conférences classe B à l'université Ibn Khaldoune de Tiaret, pour l'intérêt qu'elle a porté à ce travail en nous honorant par sa présence dans le jury.

Enfin, nos sincères remerciements vont à tous ceux qui ont contribué à la réalisation de ce travail, de près ou de loin.

## *Dédicace*

*Je dédie ce fruit à mes parents et à mes frères et sœurs.*

*Je voudrais aussi remercier ma chère épouse Oum*

*Mohamed et mes trois enfants Mohamed El Tayeb, Firas*

*Sabeur et Riad Youcef.*

*Et à tout le personnel du bureau des cartes grises de la*

*commune de Sougueur.*

*CHAKLALA ABED*

## *Dédicace*

*Je dédie ce fruit à mes parents et à mes frères et sœurs.*

*Je voudrais aussi remercier ma chère épouse Oum*

*Mohamed Amine et mes deux enfants Mohamed Amine*

*et Youcef Harroune.*

*Et à tous mes amis.*

*SADOUK BOUBAKEUR*

المبنى الذكي هو أكثر الوحدات استهلاكاً للطاقة على الإطلاق، وإذا تركت دون تحكم، فسيتم استهلاك الكثير من طاقة العالم بشكل غير مستدام. يمكن أن يحتل ظهور تقنيات إنترنت الأشياء مكاناً مهماً في عالم البناء الذكي، مما يحسن قدرته على تحويل كميات كبيرة من البيانات إلى معرفة مفيدة. اكتسبت المباني الذكية خصائص جديدة مثل: المبنى الذكي التفاعلي مع المستخدم والذي يعتمد على تطبيقات الهاتف المحمول من أجل ترشيد استهلاك الطاقة للمبنى حسب موقعه. في هذا السياق، نقترح بنية لتحسين استهلاك الطاقة للمبنى بناءً على موقع المستخدم. عادة ما يكون تمثيل الأنظمة المعقدة، مثل نظام إدارة الطاقة في المبنى، صعب التنفيذ. للتغلب على هذه المشكلة، استخدمنا نهج النمذجة الهرمية لأنظمة الأحداث المنفصلة DEVS (مواصفات نظام الأحداث المنفصلة)؛ في هذا العمل، تم تصميم النماذج الفرعية للأشياء المتصلة في المبنى بدقة وبشكل مستقل لتقييم الأنظمة الفرعية المكونة، وبعد تثبيتها معاً، يمكننا بسهولة الحصول على نموذج يخضع للتصميم المقترح. تظهر نتائج المحاكاة أن نهج التحسين الجديد هذا يحسن بشكل كبير من كفاءة الطاقة في المباني.

**الكلمات المفتاحية:** المبنى الذكي، استهلاك الطاقة، إنترنت الأشياء، موقع المستخدم.

## Résumé

Le bâtiment intelligent est l'unité la plus énergivore de toutes, et s'il n'est pas contrôlé, une grande partie de l'énergie mondiale sera consommée de manière non durable. L'avènement des technologies IdO (Internet des Objets) peut prendre une place importante dans le monde du bâtiment intelligent, améliorant sa capacité à transformer un grand nombre de données en connaissances utiles. Les bâtiments intelligents ont acquis de nouvelles caractéristiques telles que : le bâtiment intelligent interactif avec l'utilisateur qui s'appuie sur des applications mobiles afin de rationaliser la consommation énergétique du bâtiment en fonction de sa localisation. Dans ce contexte, nous proposons une architecture pour optimiser la consommation d'énergie du bâtiment en se basant sur la localisation de l'utilisateur.

Représenter des systèmes complexes, comme le système de gestion énergétique d'un bâtiment, est généralement difficile à mettre en œuvre. Pour surmonter ce problème, nous avons utilisé une approche de modélisation hiérarchique des systèmes à évènement discrets DEVS (Discrete Event System Specification) ; Dans ce travail, les sous-modèles des objets connectés dans le bâtiment sont conçus de manière précise et indépendante pour présenter les sous-systèmes composants, et après les avoir installés ensemble, nous obtenons facilement un modèle soumis à la conception proposée. Les résultats de la simulation montrent que cette nouvelle approche d'optimisation, améliore considérablement l'efficacité énergétique des bâtiments.

**Mots clés :** bâtiment intelligent, consommation d'énergie, IdO, localisation de l'utilisateur.

## Abstract

The smart building is the most energy-consuming unit of all, and if left unchecked, much of the world's energy will be consumed unsustainably. The advent of IoT (Internet of Things) technologies can take an important place in the world of smart building, improving its ability to transform large amounts of data into useful knowledge. Smart buildings have acquired new characteristics such as: the user-interactive smart building which relies on mobile applications in order to rationalize the energy consumption of the building according to its location. In this context, we propose an architecture to optimize the energy consumption of the building based on the location of the user.

Representing complex systems, such as a building's energy management system, is usually difficult to implement. To overcome this problem, we used a hierarchical modeling approach for discrete event systems DEVS (Discrete Event System Specification); In this work, the sub-models of the connected objects in the building are precisely and independently designed to present the component subsystems, and after installing them together, we easily obtain a model subject to the proposed design. The simulation results show that this new optimization approach significantly improves the energy efficiency of buildings.

Keywords: smart building, energy consumption, IoT, user location.

# Liste des tableaux

## Liste des tableaux

---

<b>Tableau I.1 Bondes des fréquences RFID .....</b>	<b>30</b>
<b>Tableau IV.1 Estimation de la consommation énergétique quotidienne de lamaison .....</b>	<b>85</b>
<b>Tableau IV.3 Données de simulation de maison.....</b>	<b>87</b>

# Liste des figures



## Liste des figures

---

Figure I.1 Liaison au monde Numérique.....	21
Figure I.2 Systèmes coopérants de gestion technique dans bâtiment.....	22
Figure I.3 Contrôle Intelligent.....	23
Figure I.4 Le bâtiment connecté .....	25
Figure I.5 Les fonctions de l'Objet connecté .....	27
Figure I.6 L'internet des Objet.....	29
Figure I.7 Réseau de Capteurs sans fils. ....	31
Figure II.1 Modélisation et simulation.....	36
Figure II.2 Un exemple de système à évènements discrets.....	37
Figure II.3 Automate à état fini.....	38
Figure II.4 modélisé par un Réseau de Petri.....	41
Figure II.5 Description du couplage Optimisation et Simulation [Poggi, 2014] .....	43
Figure II.6 Représentation graphique d'un modèle DEVS atomique.....	45
Figure II.7 Modèle atomique en action.....	46
Figure II.8 Représentation graphique d'un modèle couplé .....	47
Figure II.9 Hiérarchie de modélisation DEVS [Capoc, 2005] .....	47
Figure II.10 Correspondance entre modélisation et simulation DEVS [Capoc, 2005].....	49
Figure II.11 Un modèle DEVS atomique.....	50
Figure II.12 Un modèle DEVS couplé.....	51
Figure III.1 Politique de localisation de notre serveur local.....	56
Figure III.2 Comment localiser l'utilisateur dand notre scénario ? .....	59
Figure III.3 Flux de travail de l'économie d'énergie.....	61
Figure III.4 Système d'optimisation de la consommation d'énergie dans batiments.....	62
Figure III.5 Notation graphique d'une transition externe (a) et interne (b).....	64
Figure III.6 Système de Gestion de l'Energie dans un Bâtiment (SGEB) en DEVS.....	65
Figure III.7 Modèle atomique DEVS d'une lampe simple.....	66

## Liste des figures

---

Figure III.8	Modèle atomique DEVS d'un équipement permanent.....	68
Figure III.9	Modèle atomique d'un optimiseur .....	69
Figure III.10	Modèle atomique DEVS d'un solveur .....	71
Figure III.11	Modèle atomique DEVS d'un régulateur .....	72
Figure III.12	Modèle de simulation DEVS pour la gestion de l'énergie dans la Maison.....	74
Figure IV.1	Vue des modules logiciels JDEVS .....	78
Figure IV.2	Interface de modélisation d'un modèle couplé avec son panneau de propriétés.....	80
Figure IV.3	Composant de l'interface graphique présentant les modèles dans la bibliothèque	81
Figure IV.4	Comment effectuer des simulation dans l'environnement JDEVS .....	82
Figure IV.5	Représentation du modèle global à l'aide de l'outil JDEVS.....	83
Figure IV.6	Consommation d'énergie de la maison avec l'estimation énergétique des trois modes.....	86
Figure IV.7	Consommation d'énergie quotidienne d'éclairage de la maison .....	87
Figure IV.8	Consommation d'énergie quotidienne du réfrigérateur .....	89
Figure IV.9	Gain énergétique quotidienne de la maison.....	90
Figure IV.10	Gain énergétique hebdomadaire de la maison.....	91

# Sommaire

## Sommaire

---

Remerciement.....	2
Dédicace.....	3
Résumé.....	4
Liste de tableaux.....	7
Liste des figures.....	9
Sommaire.....	12
Introduction générale.....	16

## Chapitre I : Bâtiments intelligents et technologies modernes : Le confort à moindre cout.

I.1 Introduction .....	20
I.2 Qu'est-ce qu'un bâtiment intelligent ? .....	21
I.3 Composants du bâtiment intelligent .....	22
I.4 Le contrôle intelligent .....	23
I.4.1 le contrôle dans les bâtiments intelligents .....	24
I.4.2 le contrôle des différents composants .....	24
I.5 Le bâtiment intelligent et les objets connectés : .....	24
I.5.1 Définition d'un objet intelligent .....	26
I.5.2 Caractéristiques d'un objet intelligent .....	26
I.5.3 Les objets connectés .....	27
I.5.3.1 Les capteurs .....	27
I.5.3.2 Les sources d'énergie .....	28
I.5.3.3 Les actionneurs .....	28
I.5.3.4 La connectivité .....	28
I.6 L'Internet des Objets : un espace d'objets connectés .....	29
I.7 Les technologies clés de l'IoT et les bâtiments intelligents .....	30
I.7.1 RFID (Radio Frequency Identification) .....	30
I.7.2 WSN (Wireless Sensor Network) .....	31
I.7.3 M2M (Machine to Machine) .....	32
I.8 Conclusion .....	33

## **Chapitre II : Les systèmes à évènements discrets : La modélisation et la simulation.**

<b>II.1 Introduction</b> .....	<b>35</b>
<b>II.2 Concepts généraux</b> .....	<b>36</b>
<b>II.3 Les systèmes à évènements discrets</b> .....	<b>37</b>
<b>II.3.1 Les évènements discrets</b> .....	<b>37</b>
<b>II.3.2 Différentes approches de modélisation</b> .....	<b>38</b>
<b>II.3.2.1 Les automates à état fini</b> .....	<b>38</b>
<b>II.3.2.2 Les réseaux de Pétri</b> .....	<b>39</b>
<b>II.3.2.3 Les systèmes multi-agents</b> .....	<b>41</b>
<b>II.3.2.4 Le formalisme DEVS</b> .....	<b>42</b>
<b>II.4 Principe de couplage de l'optimisation et la simulation</b> .....	<b>43</b>
<b>II.5 La modélisation et la simulation DEVS</b> .....	<b>44</b>
<b>II.5.1 La modélisation DEVS</b> .....	<b>44</b>
<b>II.5.1.1 Le modèle atomique</b> .....	<b>45</b>
<b>II.5.1.2 Le modèle couplé</b> .....	<b>47</b>
<b>II.5.2 La simulation DEVS</b> .....	<b>49</b>
<b>II.5.3 Description d'un modèle DEVS atomique et d'un modèle DEVS couplé</b> .....	<b>50</b>
<b>II.6 Conclusion</b> .....	<b>52</b>

## **Chapitre III : Architectures des systèmes et modélisation DEVS.**

<b>III.1 Introduction</b> .....	<b>54</b>
<b>III.2 L'environnement d'application</b> .....	<b>55</b>
<b>III.2.1. Description du scénario d'application</b> .....	<b>55</b>
<b>III.2.2 Les algorithmes généraux</b> .....	<b>56</b>
<b>III.3 Contrôle du bâtiment basant sur la localisation de l'utilisateur</b> .....	<b>59</b>
<b>III.3.1. Comment localiser l'utilisateur</b> .....	<b>59</b>
<b>III.3.2. L'ajustement automatique de la politique d'économie d'énergie</b> .....	<b>60</b>
<b>III.4 Conception du système et modélisation DEVS</b> .....	<b>61</b>
<b>III.4.1. L'architecture des systèmes résultants</b> .....	<b>62</b>
<b>III.4.2. Modélisation DEVS</b> .....	<b>63</b>

III.4.2.1. La modélisation des équipements temporaires .....	66
III.4.2.2. La modélisation des équipements permanents .....	67
III.4.2.3. La modélisation de l'optimiseur .....	69
III.4.2.4. La modélisation du solveur .....	70
III.4.2.5. La modélisation du régulateur .....	72
III.4.3. Modèle de simulation DEVS .....	73
II.5.Conclusion .....	75

## Chapitre IV : Simulation et discussion des résultats.

IV.1 Introduction .....	77
IV.2 JDEVS : notre environnement de modélisation et de simulation .....	78
IV.3 Simulation du modèle proposé à l'aide de l'outil JDEVS .....	83
IV.3.1 Représentation du modèle global à l'aide de l'outil JDEVS .....	83
IV.4 Résultats expérimentaux et discussion .....	85
IV.4.1 La quantité d'énergie estimée dans différent modes de consommation.....	85
IV.4.2 Les données de simulation .....	86
IV.4.3 Un 1ier scénario : la consommation d'énergie d'un équipement temporaire.....	87
IV.4.4 2ième scénario : la consommation d'énergie d'un équipement permanent.....	88
IV.4.5 Résultats de simulation durant 24 heures .....	90
IV.4.6 Résultats de simulation durant une semaine .....	91
IV.5 Conclusion .....	91
Conclusion générale.....	94
Références.....	96

# **Introduction générale**

## **Introduction générale**

Dans le domaine des bâtiments, l'efficacité énergétique peut être définie, de manière simplifiée, comme un compromis entre l'énergie économisée et le confort des usagers, qu'il soit acoustique, thermique ou visuel. Ces dernières années, des systèmes automatisés de gestion des bâtiments de plus en plus complexes ont été étudiés dans le but de les améliorer.

Historiquement, la régulation thermique des bâtiments a été simplifiée, souvent en s'appuyant sur des thermostats, génériques ou pièce par pièce, avec un objectif de température précis. Poussées par la hausse des coûts énergétiques et la prise de conscience de l'importance de la qualité du confort, des stratégies de plus en plus avancées sont développées pour la conception des systèmes de contrôle.

Aujourd'hui, la complexité de ces derniers ne cesse de croître, notamment le contrôle de l'éclairage et l'ouverture des fenêtres. Les bâtiments sont devenus « intelligents » et intègrent désormais de véritables systèmes d'information basés sur un grand nombre de capteurs et d'actionneurs interconnectés par des réseaux de communication. La complexité de ces systèmes d'information augmente également, par exemple en permettant de détecter le comportement des utilisateurs ou en leur donnant un plus grand contrôle via des applications mobiles. Certains projets de recherche en cours visent à anticiper les usages récurrents des bâtiments en intégrant des algorithmes d'auto-apprentissage. Ces systèmes intelligents pourront déterminer l'heure à laquelle les salariés se rendent habituellement en salle de pause et ainsi activer le chauffage de la pièce avant utilisation en tenant compte de l'inertie thermique.

La gestion de l'énergie dans les bâtiments énergivores, tels que les systèmes de chauffage, de climatisation et de ventilation, est devenue une priorité de l'habitation. La recherche a été menée par des chercheurs qui ont abouti à de multiples solutions telles que la programmation informatique, l'intelligence artificielle et autres. Le système de stockage dans ce type de bâtiment est aussi une solution pour une meilleure gestion de l'énergie et des consommations locales.

L'objectif principal de ce travail est de savoir comment trouver la solution optimale de gestion de l'énergie dans un bâtiment intelligent ?



Ce domaine de recherche est impliqué dans le développement de nouvelles méthodologies visant à faciliter l'intégration, le déploiement et l'interopérabilité des capteurs et actionneurs au sein du réseau sans fil d'un bâtiment intelligent. Plus précisément, l'objectif de ce travail est de modéliser des objets connectés à un haut niveau d'abstraction après avoir défini un environnement permettant de simuler le réseau de capteurs sans fil (en anglais : WSN) faisant fonctionner le bâtiment intelligent. La Modélisation et Simulation (M&S) nous permettent de simuler différents scénarios afin d'analyser et d'évaluer divers paramètres et comportements. Ainsi, au lieu de combiner de vrais appareils fonctionnels coûteux, le concepteur configure et simule les appareils à l'aide d'un outil de modélisation et de simulation. Dans ce travail, une approche basée sur la technologie de l'IoT a été envisagée pour concevoir, par modélisation DEVS, un système de gestion de l'énergie utilisé pour planifier la production et la consommation d'énergie des appareils intelligents installés dans les bâtiments intelligents. En outre, la gestion de l'énergie dans les bâtiments intelligents s'est enrichie en tirant parti de la localisation des utilisateurs pour rationaliser leur consommation d'énergie.

Dans ce paragraphe, nous décrivons brièvement l'organisation de ce manuscrit, qui se compose de quatre chapitres. Dans le premier chapitre, nous expliquons clairement le concept de bâtiment intelligent et discutons du rôle de la technologie IoT pour garantir diverses formes de confort au moindre coût. Dans le deuxième chapitre, nous expliquons en détail le concept de systèmes à événements discrets et les différentes approches de modélisation et de simulation. Nous expliquons également tous les détails de la spécification des modèles selon le formalisme DEVS après avoir justifié notre choix de l'utiliser dans ce travail. Le troisième chapitre présente l'architecture du système de gestion de l'énergie dans les bâtiments intelligents, basé sur la localisation d'utilisateur. Cette conception a été réalisée en trois phases : l'abstraction de toutes les entités contribuant à la configuration de notre environnement applicatif, la conception d'une architecture multi-niveaux pour la gestion de l'énergie dans les bâtiments intelligents, et la spécification DEVS de tous les modèles résultants. Une fois ces trois étapes terminées, nous passons à l'étape d'implémentation et de validation des différents modèles DEVS dans différents scénarios. Ceci est décrit dans le chapitre quatre. Nous terminons ce mémoire par une conclusion générale qui retrace le contenu et les points forts de ce travail et présente les perspectives perçues.

# **CHAPITRE I**

Bâtiments intelligents et

technologie modernes :

Le confort à moindre coût.

## **Chapitre I : Bâtiments intelligents et technologies modernes :**

### **Le confort à moindre cout**

#### **I.1 Introduction**

L'énergie a un rôle majeur dans le développement économique et social et dans l'amélioration de la qualité de vie des citoyens. Si la technologie ne se développe pas et que les quantités totales n'augmentent pas de façon spectaculaire, une grande partie de l'énergie mondiale sera produite et consommée d'une manière qui ne sera pas durable à long terme. La consommation d'énergie dans les bâtiments, qu'il s'agisse de maisons, de bureaux ou d'entreprises, est un coût très élevé, surtout s'il s'agit d'une ville entière, tandis que le confort thermique, le confort visuel et la qualité de l'air de l'environnement intérieur ont un impact important sur la productivité de ses occupants. Par conséquent, les bâtiments nécessitent inévitablement une conception de haute qualité, et cela n'est possible qu'avec des études approfondies et l'utilisation de nouvelles technologies de l'information et de communication.

La transition vers un bâtiment intelligent passe par l'adoption de plateformes connectées au service des occupants et des gestionnaires. Grâce à l'Internet des Objets (en anglais: IoT), tous les composants des bâtiments peuvent désormais être connectés à moindre coût. Autrement dit, les bâtiments intelligents du futur devront faire face à deux défis; Le premier est celui du confort: le bâtiment doit rester convivial, s'adapter aux usages de ses utilisateurs et prendre en compte leurs préférences ergonomiques. Le second concerne la consommation énergétique du bâtiment. Au sein de ces deux points, une approche possible est de chercher à améliorer la manière dont les capteurs et actionneurs sont appelés à réaliser les fonctions que l'utilisateur attend des appareils de confort, afin de réduire leur consommation d'énergie nécessaire tout en assurant un tel confort.

Dans le contexte de tous ces enjeux, dans ce chapitre, nous abordons le concept de bâtiment intelligent, ses composants, le concept de confort, les objets connectés et la technologie IoT qui peut être appliquée dans cet environnement.

## I.2 Qu'est-ce qu'un bâtiment intelligent ?

Le bâtiment intelligent se définit comme un bâtiment à haute efficacité énergétique intégrant, dans la gestion intelligente du bâtiment les équipements consommateurs, les équipements producteurs et les éventuels équipements de stockage. Il s'agit de mettre de « l'intelligence » sur le réseau électrique privé des bâtiments (maison, immeuble d'habitations ou de bureaux) pour faciliter et améliorer la gestion de l'énergie et des appareils électriques sur le réseau.

Un bâtiment intelligent est « Un bâtiment qui intègre divers systèmes pour gérer efficacement les ressources dans un mode coordonné afin de maximiser les performances techniques, économie d'investissement et de fonctionnement et flexibilité, il a aussi la capacité d'apprentissage et ajustement des performances de son occupation et de l'environnement » [Panch, 2019].

Le concept de bâtiments intelligents est apparu avec l'intégration croissante de technologies avancées dans les bâtiments et leurs systèmes. Cela a donné la possibilité que tous les aspects des bâtiments puissent être exploités et contrôlés à distance pour plus de commodité, de confort et de manière rentable et économe en énergie [Jia, 2019].

Un bâtiment intelligent est un bâtiment qui cherche à répondre aux enjeux généraux de confort, de bien-être, de sécurité et de performance énergétique et environnementale par l'exploitation des technologies liées au monde numérique (figure I.1).

La gestion intelligente du bâtiment est possible grâce à la connexion en réseau des différents éléments. Sont ainsi rendus possibles :



Figure I.1 : Liaison au monde Numérique

- la transmission d'informations ;
- la gestion de données relatives à l'utilisation des différents équipements afin que le bâtiment soit en mesure d'adapter sa consommation d'énergie ;

- le stockage et l'utilisation de ces données afin d'adapter le bâtiment aux habitudes des résidents ;
- la meilleure gestion possible afin de faire un maximum d'économies d'énergie.

### I.3 Composants du bâtiment intelligent

Dans les « bâtiments intelligents », tous les équipements techniques tels que le chauffage, la climatisation, la ventilation, l'éclairage, l'ombrage, les ascenseurs et la technique de sécurité sont interconnectés et, dans la mesure du possible, commandés de façon centralisée.

Le bâtiment intelligent contient un serveur et des capteurs, en plus d'un programme qui connecte les différents capteurs dans les différents appareils électroménagers et serveurs qui conduisent à fournir un service de luxe ou à mettre la maison dans un état de service réussi.

Le serveur est l'unité de base du bâtiment intelligent, car il est comme un esprit dans le corps humain. Il contrôle tous les appareils et capteurs du bâtiment intelligent. C'est le composant qui sera généralement utilisé soit directement avec un clavier ou un écran intégré, soit indirectement avec une télécommande, soit via le WAP et un smartphone.

Les capteurs et les actionneurs du réseau qui remplacent les yeux et les oreilles du corps humain sont arrivés. Il détecte tout mouvement à l'intérieur du bâtiment intelligent et interagit avec lui, par exemple en levant les rideaux et en ajustant la température à un niveau de luxe.



Figure I.2 : Systèmes coopérants de gestion technique dans bâtiment

Au final tous les éléments communiquent entre eux. Il existe différentes solutions, comme la connexion des différents éléments avec des câbles spécialisés, considérée comme à l'abri

des interférences extérieures, mais elle est financièrement coûteuse. Depuis peu, divers appareils électroniques sont connectés au serveur via des ondes radio.

Les composants du bâtiment ci-dessus peuvent être traités sous quatre rubriques principales, à savoir [Flax, 1991] : La gestion des installations, la gestion de l'information, la connectivité et le contrôle global.

## I.4 Le contrôle intelligent

Le Contrôle Intelligent est une classe de contrôle des techniques qui utilisent diverses approches de l'intelligence artificielle comme les réseaux de neurones, la probabilité de Bayes, la logique floue, l'apprentissage automatique, l'apprentissage par renforcement, le calcul évolutif et les algorithmes génétiques [Recha, 2019].

Les fonctionnalités de contrôle intelligent permettent au hub et aux périphériques qui y sont connectés de fonctionner d'eux-mêmes. Il est possible de configurer les paramètres de contrôle intelligent en créant des "scénarios". Chaque scénario donne une réponse aux trois questions suivantes :



Figure I.3 : Contrôle Intelligent

1- **Quel** est le dispositif devant être piloté ?

- Exemple : caméra de surveillance de l'entrée principale
- Exemple : fiche intelligente de vestibule

2- **Quand** ce dispositif doit-il être mis en marche ?

- À un moment prédéterminé.
- Sur déclenchement d'un capteur.
- Au moment de rentrer à la maison.

3- **Que** doit faire le dispositif ?

- Caméra : enregistrer des images.
- Fiche intelligente : se mettre en marche.
- Thermostat : change le mode de fonctionnement actuel.

#### **I.4.1 le contrôle dans les bâtiments intelligents**

Les bâtiments intelligents ne se limitent pas aux objets de divertissement comme les télévisions intelligentes ou les enceintes connectées. Bien au contraire, cette solution technique est en constante évolution et s'avère utile pour la gestion énergétique du bâtiment.

Les bâtiments intelligents englobent les techniques et systèmes liés à l'automatisation et au contrôle d'un bâtiment. Elle peut même déceler des anomalies et proposer des solutions. L'IoT vient rattacher les installations et participe ainsi à la création d'un véritable bâtiment connecté. Les objets concernés pour la gestion énergétique du bâtiment sont aussi divers que l'éclairage, les thermostats, les fenêtres, les radiateurs et les électroménagers.

#### **I.4.2 le contrôle des différents composants**

Les composants du smart building sont surveillés par différents capteurs répartis à l'intérieur du bâtiment en communication avec le serveur à travers les réseaux disponibles (câble ou WiFi).

### **I.5 Le bâtiment intelligent et les objets connectés**

D'une manière générale, l'internet des objets (IoT) désigne l'attribution d'identifiants uniques (tels que les adresses IP) aux objets physiques. Ces identifiants permettent de se connecter à un réseau permettant le transfert de données vers et à partir de ces entités.

Ces objets peuvent être : les personnes, les animaux, les véhicules, les plantes, les appareils, les composants de bâtiments, etc., et sont souvent décrits comme des objets intelligents, par exemple un compteur intelligent, un téléphone intelligent, une maison intelligente, etc.





Ce réseau Internet ne vient pas remplacer ou se mettre à côté du réseau Internet comme nous le connaissons : il s'intègre à celui-ci en y ajoutant des objets connectés qui peuvent envoyer et/ou recevoir des informations en temps réel, agir en fonction de ces données et communiquer avec d'autres appareils. Ils utilisent pour cela les avancées technologiques qui ont permis le développement d'Internet pour pouvoir communiquer, traiter et présenter l'information

### **I.5.1 Définition d'un objet intelligent**

Selon l'Union internationale des télécommunications, l'Internet des objets (IdO) est une « infrastructure mondiale pour la société de l'information, qui permet de disposer de services évolués en interconnectant des objets (physiques ou virtuels) grâce aux technologies de l'information et de la communication interopérables existantes ou en évolution ». En réalité, la définition de ce qu'est l'Internet des objets n'est pas figée. Elle recoupe des dimensions d'ordres conceptuel et technique.

D'un point de vue conceptuel, l'Internet des objets caractérise des objets physiques connectés ayant leur propre identité numérique et capables de communiquer les uns avec les autres. Ce réseau crée en quelque sorte une passerelle entre le monde physique et le monde virtuel.

D'un point de vue technique, l'IdO consiste en l'identification numérique directe et normalisée (adresse IP, protocoles smtp, http...) d'un objet physique grâce à un système de communication sans fil qui peut être une puce RFID, Bluetooth ou Wi-Fi.

### **I.5.2 Caractéristiques d'un objet intelligent**

En peut résumer les principaux caractéristiques d'un bâtiment intelligent dans les points suivants :

- Le bâtiment maîtrise et contrôle l'espace intérieur et extérieur grâce à des systèmes automatisés permettant au bâtiment de répondre aux conditions et variables internes et externes ; tels que le changement climatique, un incendie...etc.
- Le bâtiment détermine le moyen le plus efficace et efficient et fourni un environnement pratique et confortable aux occupants.

- Le bâtiment répond aux besoins des occupants grâce à des systèmes de communication avancés.

### I.5.3 Les objets connectés

L'objet connecté est d'abord un objet qui a une fonction mécanique et/ou électrique propre, il peut soit être conçu directement « connectable », soit il est déjà existant et la connectivité est rajoutée à postériori.

L'objet connecté a pour fonction de collecter des données de capteurs, de traiter ces données et de les communiquer à l'aide de d'une fonction de connectivité et de recevoir des instructions pour exécuter une action. Généralement ces fonctions de l'objet connecté nécessitent une source d'énergie, surtout quand les données sont prétraitées directement dans l'objet.

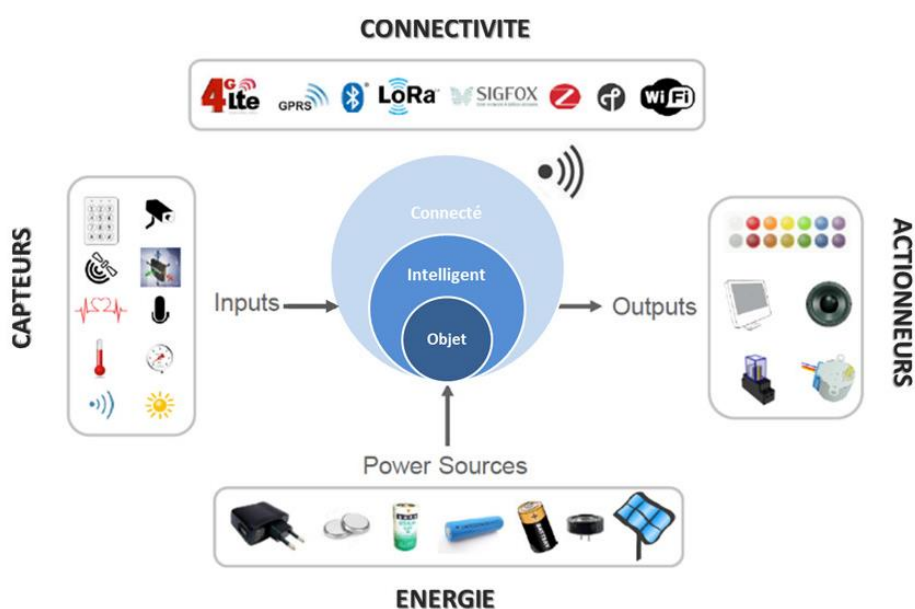


Figure I.5 : Les fonctions de l'Objet connecté [Conne, 2021]

#### I.5.3.1 Les capteurs

Dans de nombreux domaines (industrie, recherche scientifique, services, loisirs...), on a besoin de contrôler des paramètres physiques (température, force, position, vitesse, luminosité...). Le capteur est l'élément indispensable à la détection de ces grandeurs physiques [Conne, 2021].

Un capteur est un organe de prélèvement d'informations qui élabore à partir d'une grandeur physique, une autre de nature différente (souvent électrique).

Cette grandeur représentative de la grandeur prélevée est utilisable à des fins de mesure ou de commande.

On peut classer les capteurs en 3 groupes en fonction de l'information délivrée en sortie :

- A) Les capteurs analogiques : Dans la pratique industrielle, on donne à ce type de matériel le nom de capteurs. Type de signal de sortie : 0 – 10 V ou 4 – 20 mA ;
- B) Les capteurs numériques : Souvent nommés codeurs ou compteurs. Type de signal de sortie : 0011 ou 0001 ;
- C) Les capteurs logiques ou Tout Ou Rien (TOR) : Ils portent le nom de détecteurs. Type de signal de sortie 0 V ou 5 V.

### **I.5.3.2 Les sources d'énergie**

Le terme source d'énergie désigne tous les phénomènes, physiques et chimiques, qui permettent d'obtenir de l'énergie. Les sources d'énergie sont nombreuses à être exploitées : elles sont utilisées à des fins variées, et cela dans différents domaines de la vie quotidienne. Certaines sont connues depuis toujours, d'autres bénéficient d'une découverte plus récente [Conne, 2021].

Ces sources d'énergie sont ensuite classées en deux catégories, les sources d'énergie renouvelables et les sources d'énergie non renouvelables.

### **I.5.3.3 Les actionneurs**

Un actionneur est un dispositif qui convertit un signal électrique en sortie physique [Joche, 2018], c'est-à-dire un mouvement. Un actionneur peut être contrôlé par la tension ou le courant électrique, la pression pneumatique ou hydraulique [Julio, 2013] ou même la puissance humaine. Dans les systèmes embarqués, les actionneurs sont principalement contrôlés par l'électricité. Lorsque le signal de commande est reçu, l'actionneur convertit l'énergie électrique en mouvement mécanique. Les actionneurs peuvent créer un mouvement linéaire, un mouvement rotatif ou un mouvement oscillatoire. Des exemples d'actionneurs comprennent les moteurs électriques, les actionneurs piézoélectriques, les actionneurs pneumatiques, les moteurs pas à pas et les actionneurs de serrure de porte, etc [Richa, 2006].

### **I.5.3.4 La connectivité**

La connectivité de l'objet est assurée par une petite antenne RF (Radio Fréquence) qui permettra la communication de l'objet vers un ou plusieurs réseaux.

D'une part, les objets peuvent rapporter des informations telles que leur identité, leur état, des alarmes ou des données de capteur, et d'autre part peuvent recevoir des informations telles que des commandes d'action et des données. Le module de connexion peut également gérer le "cycle de vie des objets", c'est-à-dire l'authentification et l'enregistrement, le débogage, la mise à jour et la suppression des objets connectés dans le réseau [Maato, 2021].

## I.6 L'Internet des Objets : un espace d'objets connectés

L'Internet des objets (ou Internet of Things en anglais, IoT) est une infrastructure mondiale pour la société de l'information, permettant des services avancés en interconnectant des objets (physiques et virtuels) basés sur des technologies de l'information et de la communication interopérables existantes et en évolution [Ray, 2016].

La définition commune de l'Internet des objets est : Internet des objets est un réseau d'objets physiques sur internet est non seulement un réseau d'ordinateurs, mais il est devenu un réseau d'appareils de tous types et tailles, véhicules, smart téléphones, appareils électroménagers, appareils photo, instruments médicaux et les systèmes industriels, les bâtiments, tous connectés, toutes les informations de communication et de partage basés sur protocoles stipulés afin de réaliser des réorganisations intelligentes, positionnement, traçage, sécurité et contrôle et même en temps réel personnel surveillance en ligne, mise à niveau en ligne, contrôle des processus et administration [Patel, 2016].



Figure I.6 : L'internet des Objet

Selon l'Union internationale des télécommunications, l'Internet des objets (IdO) est une « infrastructure mondiale pour la société de l'information, qui permet de disposer de services évolués en interconnectant des objets (physiques ou virtuels) grâce aux technologies de l'information et de la communication interopérables existantes ou en évolution ».

D'un point de vue conceptuel, l'Internet des objets caractérise des objets physiques connectés ayant leur propre identité numérique et capables de communiquer les uns avec les autres. Ce réseau crée en quelque sorte une passerelle entre le monde physique et le monde virtuel.

D'un point de vue technique, l'IdO consiste en l'identification numérique directe et normalisée (adresse IP, protocoles SMTP, http...) d'un objet physique grâce à un système de communication sans fil qui peut être une puce RFID, Bluetooth ou Wi-Fi.

## I.7 Les technologies clés de l'IoT et les bâtiments intelligents

Infrastructure mondiale pour la société de l'information qui permet de disposer de services évolués en interconnectant des objets (physiques ou virtuels) grâce aux technologies de l'information et de la communication interopérables existants ou en évolution (UIT)<sup>1</sup>.

### I.7.1 RFID (Radio Frequency Identification)

RFID est l'acronyme de "Radio Fréquence Identification". La technologie RFID présente de nombreux avantages. Elle permet de scanner des produits en masse sans avoir à les voir, sans avoir

Band	Range	Data Speed	Tags
Low frequency (LF) : 125–134.2 kHz	10 m	Low	Passive
High frequency (HF) : 13.56 MHz	10 cm–1 m	Low to moderate	Passive
Ultra high frequency (UHF) : 433 MHz	1–100 m	Moderate	passive or active
Ultra high frequency (UHF) : 856 MHz–960 MHz	1–12 m	moderate to high	passive or active
Microwave : 2.45–5.8 GHz	1–2 m	High	Active
Microwave : 3.1–10 GHz	<200 m	High	Active

Tableau I.1 : Bondes des fréquences RFID

<sup>1</sup> UIT : l'Union Internationale des Télécommunications

à les toucher et donc permet un gain de temps considérable. Il s'agit d'une méthode utilisée pour transférer les données des étiquettes à des objets, ou pour identifier les objets à distance (l'étiquette contient des informations stockées électroniquement pouvant être lues à distance) [Maato, 2021].

L'étiquette RFID se compose principalement d'une puce électronique et d'une antenne bobinée ou imprimée. L'antenne, souvent composée de cuivre, est déposée sur l'étiquette grâce à des ultrasons (systèmes de vibrations). La taille de la puce, quant à elle, est variable selon les applications. Elle peut désormais être réduite à celle d'un point comme les puces développées en 2008 par HITACHI de type « poudre » qui mesure seulement 0,05 mm x 0,05 mm.

Un système RFID comprend généralement une étiquette, un lecteur et une antenne. Le lecteur envoie un signal d'interrogation à l'étiquette via l'antenne, et l'étiquette répond avec ses informations uniques. Les étiquettes RFID peuvent être actives ou passives. Les étiquettes RFID actives ont leur source d'alimentation et peuvent donc être lues sur une longue portée (jusqu'à 100 mètres) [Sara, 2014]. Les étiquettes RFID passives n'ont pas leur source d'alimentation. Ils sont alimentés par l'énergie électromagnétique transmise par le lecteur RFID. Par conséquent, ils ne peuvent être lus que sur une courte distance (<25m) [Xiaol, 2012] La RFID fonctionne principalement dans les gammes de fréquences suivantes, comme indiqué dans Tableau 1.

### I.7.2 WSN (Wireless Sensor Network)

C'est un ensemble de nœuds qui communiquent sans fil et qui sont organisés en un réseau coopératif. Chaque nœud possède une capacité de traitement et peut contenir différents types de

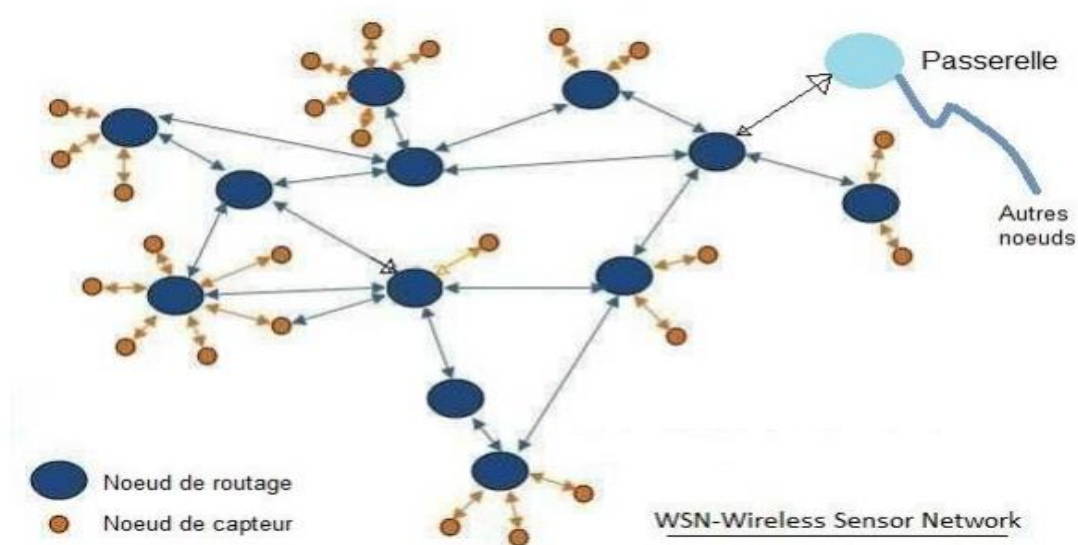


Figure I.7 : Réseau de Capteurs sans fils

mémoires, un émetteur-récepteur RF et une source d'alimentation, comme il peut aussi tenir compte des divers capteurs et des actionneurs. Comme son nom l'indique, le WSN constitue alors un réseau de capteurs sans fil qui peut être une technologie nécessaire au fonctionnement de l'IoT.

### **I.7.3 M2M (Machine to Machine)**

Machine-to-Machine (M2M) représente un futur IoT où des milliards d'objets quotidiens et des environnements environnants sont connectés et gérés à travers une gamme de dispositifs, de réseaux de communication et de serveurs en Cloud [Bandy, 2013]. La communication M2M, également appelée MTC, permet d'échanger des informations entre machines et machines sans interaction humaine. En effet, la communication M2M est une nouvelle technologie de communication par laquelle un grand nombre de "dispositifs intelligents" peuvent communiquer entre eux de façon autonome et prendre des décisions collaboratives sans intervention humaine directe [Colit, 2011] pour améliorer la rentabilité et la gestion du temps. La communication Machine-to-Machine a son origine dans les systèmes de contrôle de supervision et d'acquisition de données (SCADA), où des capteurs et d'autres appareils connectés par câble ou sans fil sont utilisés avec des ordinateurs pour surveiller et contrôler les processus industriels..

Les communications machine-to-machine sont définies comme l'échange d'informations entre machines et machines sans interaction humaine. Avec l'interconnexion à internet, un grand nombre de dispositifs sont organisés de manière autonome pour constituer un réseau M2M. Les réseaux M2M devraient être largement utilisés dans de nombreux domaines des applications omniprésentes [Decui, 2010] notamment l'automatisation industrielle et agricole, les soins de santé, les systèmes de transport, les réseaux électriques, etc. Deux principales caractéristiques des réseaux M2M :

- 1) Un nombre énorme de dispositifs dans la couverture de service et la tentative d'accès réseau simultané à partir de ces dispositifs.
- 2) Haut niveau d'automatisation du système dans lequel les dispositifs et les systèmes peuvent échanger et partager des données.

Par conséquent, la gestion massive des accès et le protocole d'accès au support sont les principaux problèmes des communications M2M pour la mise en place d'un système de communication évolutif, flexible et automatique [Wu, 2011].

## **I.8 Conclusion**

Dans ce chapitre, nous avons présenté un concept général concernant les bâtiments intelligents et le bien-être qu'ils procurent à leurs résidents, en tenant compte de la rationalisation de la consommation d'énergie dans les bâtiments intelligents. Nous avons également introduit des technologies modernes qui facilitent le processus de rationalisation de la consommation d'énergie, en particulier l'Internet des objets et certaines de ses applications.

Le développement des technologies de l'information s'est manifesté dans les bâtiments intelligents, ce qui a entraîné la modification de l'environnement quotidien des résidents des bâtiments intelligents, à mesure que le nombre d'appareils numériques augmentait et que leur utilisation en interaction avec l'utilisateur. Nous avons également présenté les différentes technologies mises en place par des designers tels que FRID et WSN et M2M dans le domaine des bâtiments intelligents.



# **CHAPITRE II**

Les systèmes à événements  
discrets : La modélisation et  
simulation.

## **Chapitre II : Les systèmes à évènements discrets : La modélisation et simulation.**

### **II.1 Introduction**

Dans le chapitre précédent, nous avons expliqué qu'est-ce qu'un bâtiment intelligent, et les composants qui le traduisent. Dans ce chapitre L'étude s'est ensuite orientée vers la modélisation. Et la simulation du bâtiment et de ses actionneurs pour prédire son comportement et ainsi alimenter le contrôle. En utilisant les outils offerts par les mathématiques, la physique et l'informatique, nous avons donc développé des modèles numériques qui tentent de prédire le comportement du bâtiment soumis à différentes simulation.

Dans la première partie de ce chapitre, nous présenterons des concepts généraux avec une définition brève des différentes étapes de modélisation utilisées pour prédire le comportement d'un bâtiment. Dans la deuxième partie nous définiront qu'est qu'un SED et les techniques pour le traduire afin de définir une architecture de modélisation. Ensuite, dans la troisième partie, nous nous pencherons sur Le formalisme DEVS et leurs optimisations.

## II.2 Concepts généraux

La modélisation et la simulation trouvent leur origine dans divers domaines allant de la théorie systémique, l'analyse numérique, les sciences informatiques, les mathématiques et la physique.

Comme nous pouvons le voir sur figure II.1, modélisation et simulation sont deux processus permettant la manipulation de trois entités [Zeigl, 1976] (Figure II.1) : « le système réel », « le modèle », « le simulateur » que nous allons définir.

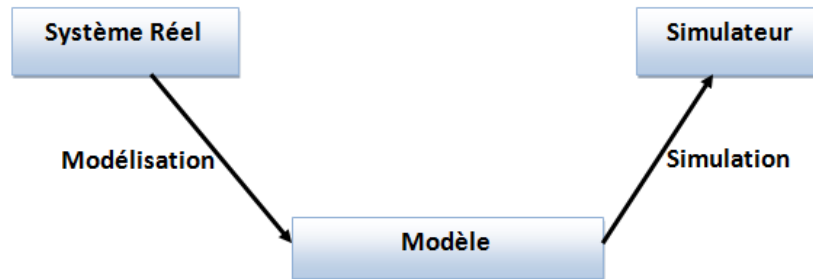


Figure II.1 : Modélisation et Simulation [Zeigl, 1976]

Le système réel définit une partie observée du monde réel [Cassa, 1999]. C'est un phénomène observé dans un environnement donné. L'environnement spécifie les conditions dans lesquelles le système fonctionnera et permet son expérimentation et sa validation. Tout système réel peut être représenté sous la forme d'un modèle suite à une phase de modélisation. La modélisation est un processus d'analyse et de structuration formelle du système réel en vue de créer une représentation simplifiée [Fishm, 2001] : « le modèle ». Il peut se présenter sous une forme numérique ou symbolique.

Le « modèle » est une représentation du problème résultant du processus de modélisation [Leemi, 2006] C'est la représentation du système basée généralement sur la définition d'ensemble d'instructions, de règles, d'équations, de contraintes, permettant de générer un comportement après simulation. Sa qualité est estimée à partir de la comparaison entre les données qu'il génère lors de sa simulation et les données observables sur le système réel.

Un « simulateur » permet de modéliser un système réel. Il présente donc, sous des conditions contrôlables et observables, l'évolution du modèle du phénomène. En d'autres termes, c'est une entité qui est responsable de l'interprétation du modèle (exécuter ces instructions) pour générer son comportement. La simulation est donc le processus d'exécution du comportement du modèle. Elle permet l'évolution de son état dans le temps en fonction d'un contexte spécifique : « le scénario ».



## II.3.2 Différentes approches de modélisation

### II.3.2.1 Les automates à état fini

Un automate est une machine qui a des entrées et des sorties discrètes, et qui réagit à une modification de ses entrées en changeant ses sorties. Cette machine est constituée d'états et de transitions associées à des événements. Le changement de sortie (passer d'un état à un autre état) dépend du mot (séquence) fourni en entrée.

Les automates à états finis sont des automates dont le nombre d'états est fini. Il est dit *déterministe* dans le sens où depuis tout état, il n'existe pas plusieurs arcs de sortie qui soient associées à un même symbole et qui conduisent à des états différents.

Considérons le diagramme états-transitions de Figure II- 3, où les nœuds représentent les états et les arcs associés à des symboles de l'alphabet représentent les transitions entre les états. L'ensemble des états  $Q = \{ q_0, q_1, q_2, q_3 \}$ . L'état initial  $q_0$  est représenté par un arc entrant. Les états finaux (ou marqués) sont représentés par des doubles cercles, ici :  $Q_m = \{ q_0, q_2 \}$ . L'ensemble des symboles associés aux arcs est l'ensemble des événements (alphabet)  $\Sigma = \{ a, b, c, d \}$ .

Les arcs dans ce diagramme modélisent une représentation graphique de la fonction de transition  $\delta : Q \times \Sigma \rightarrow Q$ :

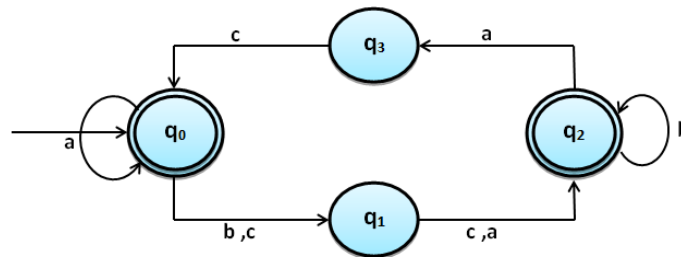


Figure II.3 : Automate à état fini

$$\delta(q_0, a) = q_0 \quad \delta(q_0, b) = \delta(q_0, c) = q_1$$

$$\delta(q_1, d) = q_1 \quad \delta(q_1, c) = \delta(q_1, a) = q_2$$

$$\delta(q_2, b) = q_2 \quad \delta(q_2, a) = q_3$$

$$\delta(q_3, c) = q_0$$

On peut remarquer que :

- L'occurrence d'un événement ne change pas forcément d'état  $\delta(q_0, a) = q_0$ .
- A partir d'un état, on peut avoir deux événements différents dont l'occurrence mène à un

même état ;  $\delta(q_0, b) = \delta(q_0, c) = q_1$ .

- La fonction  $\delta$  est une fonction partielle dans son domaine  $Q \times \Sigma$  puisqu'on n'a pas besoin d'avoir une transition pour chaque événement de  $\Sigma$  à partir de chaque état ; par exemple les instances  $\delta(q_0, d)$  n'sont pas définies.

Un automate  $A$  est un quintuple  $A = (Q, \Sigma, \delta, q_0, Q_m)$ , peut être formellement défini comme suit:

- $Q$  est l'ensemble fini des états.
- $\Sigma$  est un ensemble fini de symboles d'entrée (ensemble des événements), appelé alphabet d'entrée.
- $\delta$  est la fonction de transition d'états de  $Q \times \Sigma$  vers  $Q$  qui associe un état d'arrivée à un état de départ et à un symbole d'entrée.
- $q_0 \in Q$  est l'état initial.
- $Q_m \subseteq Q$  est l'ensemble des états marqués.

**Définition 1.1 :** Un état  $q \in Q$  est dit accessible s'il existe une chaîne  $s \in \Sigma^*$  telle que  $q = \delta(q_0, s)$ . Par extension, un automate  $A$  est dit accessible si tous ses états sont accessibles.

**Définition 1.2 :** Un état  $q \in Q$  est dit co-accessible s'il existe une chaîne  $s \in \Sigma^*$  telle que  $(q_0, s) \in Q_m$ . Par extension, un automate  $A$  est dit co-accessible si tous ses états sont co-accessibles.

**Définition 1.3 :** Le langage généré par un automate  $A = (Q, \Sigma, \delta, q_0, Q_m)$  est

$$L(A) = \{s \in \Sigma^* \mid \delta(q_0, s) \text{ est défini}\}$$

Le langage marqué par l'automate  $A$  est

$$L_m(A) = \{s \in \Sigma^* \mid \delta(q_0, s) \in Q_m\}$$

Le langage généré par un automate est toujours préfixe-clos  $L(A) = L(A)$ .

Le langage  $L_m(A) \subseteq L(A)$  puisque  $Q_m \subseteq Q$ .

**Définition 1.4 :** Les deux automates  $A1$  et  $A2$  sont dits équivalents si

$$L(A1) = L(A2) \text{ et } L_m(A1) = L_m(A2).$$

### II.3.2.2 Les réseaux de Pétri

Dès leur création en 1962 par Carl Adam Petri, les réseaux de Petri constituent un outil très approprié pour étudier les systèmes à événements discrets en raison de la puissance de modélisation et de leurs propriétés mathématiques. Les modèles obtenus, leur représentation graphique de la structure du système étudié, permettent une analyse des propriétés et par

conséquent une validation de leurs spécifications.

Les réseaux de Petri ont, par rapport aux automates, l'avantage d'être un modèle beaucoup plus général, bénéficiant de structures beaucoup plus riches, s'adaptant parfaitement à la description de différents types de SED. Les réseaux de Petri (RdP) possèdent un intérêt fondamental indéniable puisqu'ils permettent de modéliser et de maîtriser les comportements des systèmes parallèles et distribués, synchronisés et communicants.

Par leur relation avec les machines à états, les réseaux de Petri constituent des représentations faciles à comprendre et donc à manipuler, à la fois pour la création des modèles et pour leur analyse.

### **Notions de Base :**

Un Réseau de Petri est un graphe orienté composé de deux types de nœuds, de places et de transitions. Des arcs relient certaines places à certaines transitions ou certaines transitions à certaines places. Un arc ne peut pas relier deux nœuds de même type.

L'ensemble fini de places,  $P = \{p1, p2, p3, \dots, pm\}$ , symbolisées par des cercles et représentant des conditions :

- Une ressource du système (ex. : une machine, un stock, un convoyeur...).
  - L'état d'une ressource du système (ex. : machine libre, stock vide, convoyeur en panne...).
  - L'ensemble fini de transitions,  $T = \{t1, t2, t3, \dots, tn\}$ , symbolisées par des traits et représentant l'ensemble des événements (les actions se déroulant dans le système) dont l'occurrence provoque la modification de l'état du système.
- Un ensemble fini d'événements associés à chaque transition.
- Un ensemble fini d'arcs orientés qui assurent la liaison d'une place vers une transition ou d'une transition vers une place.

À l'intérieur des **places**, un nombre (positif ou nul) de **marques** ou **jetons** peut indiquer : par exemple, une condition reliée à cette place est vraie (place marquée) ou fausse, ou la quantité de ressource en stock qui est représenté par cette place, etc.

L'état d'un RdP est donné par le nombre de jetons dans chaque place.

**Définition 1.5 :** De manière plus formelle, un RdP peut être défini comme un 4-uplet

$(P, T, Post, Pre)$  :

- $P$  et  $T$  sont respectivement l'ensemble fini et non vide de places et de transitions.
- $Post$  est l'application d'incidence arrière,  $Post : (T \times P) \rightarrow \mathbb{N}^+$ .
- $Pre$  est l'application d'incidence avant,  $Pre : (P \times T) \rightarrow \mathbb{N}^+$ .

Un arc est de la forme  $(p_i, t_j)$  ou  $(t_j, p_i)$ , et le poids relatif à un arc est un entier positif. Si les poids des arcs sont tous égaux à 1, alors le réseau de Petri est dit ordinaire. Si l'arc  $(p, t) \in A$ , alors on dit que  $p$  (respectivement  $t$ ) est une place d'entrée de la transition  $t$  (respectivement transition de sortie de la place  $p$ ). Si une place  $p$  est à la fois place d'entrée et de sortie d'une même transition  $t$ , alors elle est dite impure, sinon elle est pure. Un RdP pur est un RdP qui ne contient aucune place impure, sinon il est impur. Dans la suite, nous utiliserons les notations suivantes :

- $p^\circ$  (respectivement  ${}^\circ p$ ) désigne l'ensemble des transitions de sortie (respectivement d'entrée) de la place  $p$ .
- $t^\circ$  (respectivement  ${}^\circ t$ ) désigne l'ensemble des places de sortie (respectivement d'entrée) de la transition  $t$ .

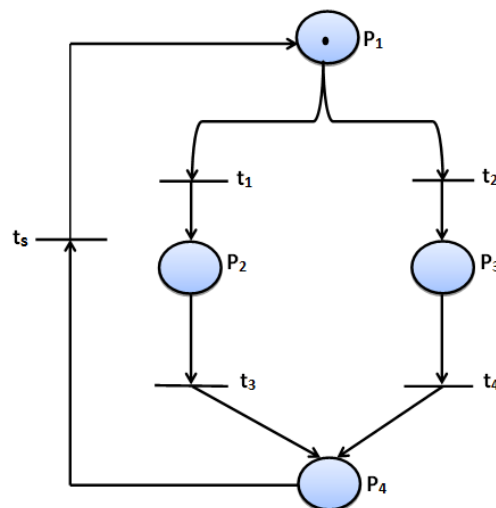


Figure II.4 : Modélisé par un Réseau de Petri

### II.3.2.3 Les systèmes multi-agents

Pour Weiss (1999), un **agent** est une "entité computationnelle", comme un programme informatique ou un robot, qui peut être vue comme percevant et agissant de façon autonome sur son environnement. On peut parler d'autonomie parce que son comportement dépend au moins partiellement de son expérience. Un système multi-agents (SMA) est constitué d'un ensemble de processus informatiques se déroulant en même temps, donc de plusieurs agents vivant au même moment, partageant des ressources communes et communiquant entre eux. Le point clé des



systèmes multi-agents réside dans la formalisation de la **coordination** entre les agents. La recherche sur les agents est ainsi une recherche sûre :

- **La décision** -quels sont les mécanismes de la décision de l'agent ? Quelle est la relation entre les perceptions, les représentations et les actions des agents ? Comment décomposent-ils leurs buts et tâches ? Comment construisent-ils leurs représentations ?
- **Le contrôle** - quelles sont les relations entre les agents ? Comment sont-ils coordonnés ? Cette coordination peut être décrite comme une coopération pour accomplir une tâche commune ou comme une négociation entre des agents ayant des intérêts différents.
- **La communication** - quels types de message s'envoient-ils ? à quelle syntaxe obéissent ces messages ? Différents protocoles sont proposés en fonction du type de coordination entre les agents.

Les systèmes multi-agents ont des applications dans le domaine de l'intelligence artificielle où ils permettent de réduire la complexité de la résolution d'un problème en divisant le savoir nécessaire en sous-ensembles, en associant un agent intelligent indépendant à chacun de ces sous-ensembles et en coordonnant l'activité de ces agents [Erber, 1995]. On parle ainsi d'intelligence artificielle distribuée. Cette méthode s'applique, par exemple, pour la surveillance d'un processus industriel ou où elle met en œuvre la solution de bon sens qui consiste à coordonner plusieurs surveillants spécialisés, plutôt qu'à envisager un seul surveillant omniscient.

Les recherches fondamentales concernent la représentation de la décision des agents, ou les protocoles de communication. Elles s'appliquent principalement aux télécommunications, à l'Internet avec le commerce électronique, à des agents physiques tels les robots, à l'optimisation des systèmes de transports et à la gestion de filières.

#### II.3.2.4 Le formalisme DEVS

Enfin un dernier formalisme nommé « Discrete Event System Specification » (DEVS) [Zeigl, 1993] permet également la modélisation et la simulation de systèmes à évènements discrets. Il propose une méthode mathématiquement formelle. Parmi les différents formalismes énoncés, le formalisme DEVS est celui qui offre la représentation des modèles la plus intuitive. En effet les systèmes modélisés peuvent être traduits facilement vers une représentation graphique et hiérarchique sous forme de boîtes noires interconnectées et imbriquées. Cette représentation permet de faciliter les échanges entre le modélisateur et les spécialistes du système étudié. Fortement inspiré par les concepts de la programmation orientée objet, il répond ainsi à nos besoins

de généricité, de compatibilité et de réutilisabilité.

De plus, le formalisme DEVS peut aujourd'hui être considéré comme un « formalisme multi paradigme ». Grâce à différentes extensions et intégrations, celui-ci englobe d'autres formalismes tels que les équations différentielles [D'abre, 2003] pour la simulation de systèmes continue encore les réseaux de Pétri [Bouke, 2013] pour la simulation de systèmes à événements discrets. Il se positionne à un niveau d'abstraction supérieur à celui de ses concurrents [Vangh, 2002] et permet d'effectuer des transformations d'un formalisme à un autre [Lara, 2002]. Enfin, il offre aussi la possibilité de réaliser des simulations de systèmes hétérogènes.

Toutes ces qualités nous ont conduits à choisir le formalisme DEVS pour le développement de notre outil d'aide à la décision. Celui-ci est présenté de manière approfondie dans la partie suivante.

## II.4 Principe de couplage de l'optimisation et la simulation

Plusieurs méthodes d'optimisation ont été appliquées aux problèmes dans lesquels la M&S est utilisé. Cet attrait entre les deux concepts s'explique par la possibilité de reformuler les modèles et par la possibilité offerte par la simulation d'étudier de nombreux scénarios et ces deux caractéristiques contribuent à une meilleure optimisation. L'objectif est de trouver les solutions optimales en simulant le problème à optimiser.

Chacun des deux domaines possède son propre sens. Comme nous voyons sur la figure II.5, du point de vue de l'optimisation, les données émises sont des « solutions », les données reçues des

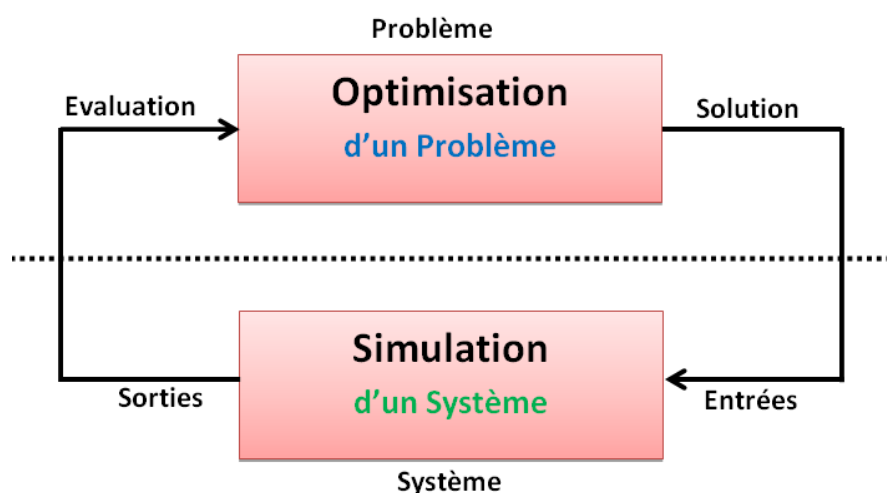


Figure II.5 : Description du couplage Optimisation et Simulation [Poggi, 2014]

« évaluations » et l'entité manipulée le « problème ». Du point de vue de la M&S, les données reçues sont des « entrées », les données générées sont des « sorties » et l'entité traitée est le « système ». Récemment, plusieurs travaux se sont appuyés sur l'intégration de l'optimisation dans la M&S dans leurs méthodologies. Ces différentes approches posent les bases du couplage de ces deux domaines. Les résultats obtenus sont encourageants et démontrent l'intérêt de l'intégration des deux domaines. Encore une fois, compte tenu de ce qui précède, les qualités du formalisme DEVS dans le domaine « la M&S de systèmes à évènements discrets », et sa grande capacité à s'intégrer dans le domaine de « l'optimisation de problèmes complexes », tout cela nous a poussés à le choisir pour développer notre modèle général de la manière optimale que nous cherchons.

## II.5 La modélisation et la simulation DEVS

Le formalisme DEVS<sup>2</sup> [Zeigl, 1976] a été introduit à la fin des années 70 par le Professeur B.P Zeigler qui avait pour objectif de donner un socle mathématique solide à la M&S des systèmes à événement discrets. Un système à évènements discrets est un système pouvant être décrit à partir d'un ensemble d'états et de règles de transition entre ces états.

Le formalisme DEVS permet la représentation d'un système comme un modèle ou un ensemble de modèles possédant des états et des transitions. De plus, il offre la possibilité de modéliser et de simuler des modèles informatiques et mathématiques dans un cadre formel [Guyom, 2011]. Le formalisme se base sur un ensemble de spécifications. Il peut être considéré comme une approche standard et universelle de modélisation et de simulation [Vangh, 2002] dont le principal avantage est la séparation explicite entre la modélisation du système, son simulation et son analyse.

### II.5.1 La modélisation DEVS

La modélisation est un processus d'analyse et de structuration formelle du système réel en vue de créer une représentation simplifiée [Fishm, 2001] : « le modèle ». La modélisation DEVS s'articule autour de deux catégories de modèles : les « modèles atomiques » pour l'aspect comportemental des systèmes et les « modèles couplés » pour l'aspect structurel.

Par définition, un modèle couplé peut contenir plusieurs modèles atomiques ou plusieurs modèles couplés. Par contre un modèle atomique est indissociable et tout modèle couplé peut être représenté par un modèle atomique unique.

---

<sup>2</sup>Discrete Event system Specification

Le modèle atomique est la pièce élémentaire de la modélisation DEVS et il est défini par un ensemble d'états et de fonctions de transition entre les états de l'automate [Capoc, 2014].

### II.5.1.1 Le modèle atomique

Le formalisme de base DEVS dit « classique » considère un modèle atomique comme un concept mathématique basé sur le temps, un ensemble de valeur caractérisant tous les stimuli possibles en entrée et en sortie du système (voir figure II.6) ainsi que deux fonctions permettant de déterminer la réponse comportementale à ces stimuli [Capoc, 2005].

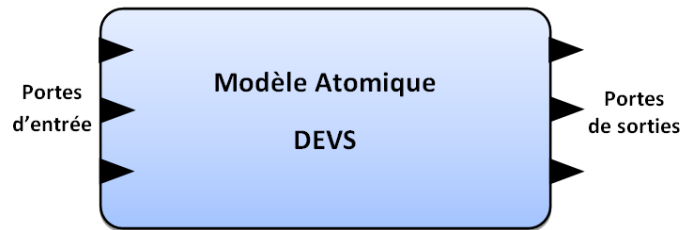


Figure II.6 : Représentation graphique d'un modèle DEVS atomique

#### A- Spécification DEVS atomique

Dans le formalisme DEVS classique avec port, un modèle atomique MA est spécifié par la définition de sept entités :

$$MA = \langle X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta \rangle \quad \text{avec :}$$

1. l'ensemble des entrées :  $X = \{(p,v) | p \in P_e, v \in V_x\}$  où  $P_e$  et  $V_x$  sont deux ensembles finis représentant l'ensemble des ports d'entrées et des valeurs portées par les événements reçus en entrée ;
2. l'ensemble des sorties :  $Y = \{(p,v) | p \in P_s, v \in V_y\}$  où  $P_s$  et  $V_y$  sont deux ensembles finis représentant l'ensemble des ports de sorties et des valeurs portées par les événements générées en sortie ;
3. l'ensemble des états :  $S = \{s_i | i \in \mathbb{R}^+\}$  ;
4. la fonction de transition interne :  $\delta_{int} (S) \rightarrow S$  ;
5. la fonction de transition externe :  $\delta_{ext} (Q \times X) \rightarrow S$  ;
6. la fonction de sortie :  $\lambda (S) \rightarrow Y$  ;
7. la fonction d'avancement du temps :  $ta (S) \rightarrow \mathbb{R}^+$  (réels positifs de 0 à  $\infty$ ).

Les modèles réagissent à deux types d'événements : les événements externes et les événements internes. Les événements externes proviennent d'un autre modèle, déclenchent la fonction de transition externe et mettent à jour le temps de vie du composant. Les événements internes correspondent à des changements d'états du modèle, déclenchent les fonctions de transitions internes et de sorties, le modèle calcule ensuite grâce à la fonction  $ta$  la date du prochain événement interne.

### B- Description comportementale

L'interprétation de ces éléments est illustrée sur la figure II.7. A chaque instant, le système (modèle atomique) est dans un état  $s$ . Si aucun événement externe n'intervient, le système restera dans cet état pendant un temps donné par la fonction  $ta(s)$ . Lorsque le temps de vie du modèle est expiré, c'est-à-dire lorsqu'il s'est écoulé  $e = ta(s)$ , le système active sa fonction de sortie  $\lambda(s)$  et change d'état grâce à l'exécution de la fonction de transition interne  $\delta_{int}(s)$ .

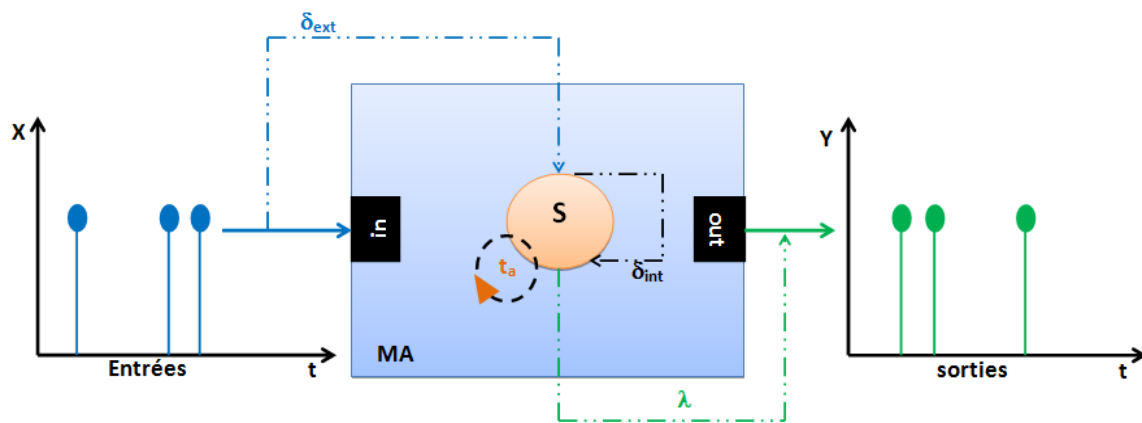


Figure II.7 : Modèle atomique en action

Si un événement externe  $x \in X$  intervient avant que le temps ne soit expiré, i.e. quand le système est dans l'état total  $(s,e)$  avec  $e \leq ta(s)$ , le système change d'état grâce à l'exécution de la fonction de transition externe  $\delta_{ext}(s,e,x)$ . Dans les deux cas, le système est alors dans un nouvel état  $s'$  avec un nouveau temps restant  $ta(s')$  et ainsi de suite.

Le temps de vie du composant peut être égal à  $0$  ou à  $\infty$ . Dans le premier cas, la durée de l'état  $s$  est tellement courte qu'aucun événement externe ne peut intervenir avant l'arrivée du prochain changement d'état. Nous disons de  $s$  qu'il est dans un état transitoire. Dans le second cas, le système restera dans l'état  $s$  indéfiniment si aucun événement externe ne vient l'interrompre. Nous disons de  $s$  dans ce cas qu'il est dans un état passif.

### II.5.1.2 Le modèle couplé

Le formalisme DEVS utilise la notion de hiérarchie de description qui permet la construction de modèles dits « couplés » à partir d'un ensemble de sous-modèles et de trois relations de couplage avec ces sous-modèles [Capoc, 2005].

Un exemple graphique de modèle couplé est donné au niveau de la figure II.8. On peut y remarquer la construction d'un modèle couplé à partir de deux modèles A et B qui peuvent être également des modèles atomiques ou couplés. L'essentiel de la définition consiste à diriger les événements entrants, sortants et internes aux bons endroits.

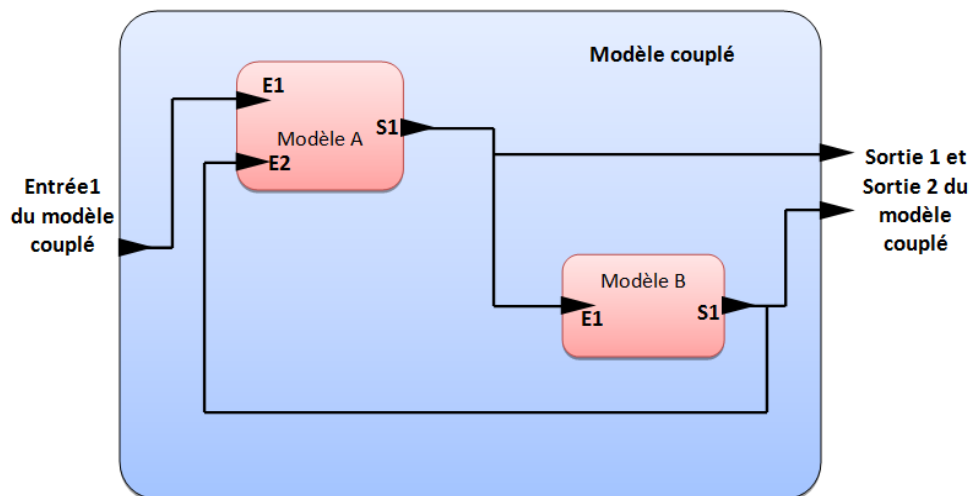


Figure II.8 : Représentation graphique d'un modèle couplé

Un modèle couplé possède les trois relations de couplage suivantes [Capoc, 2005] (voir figure II.9) :

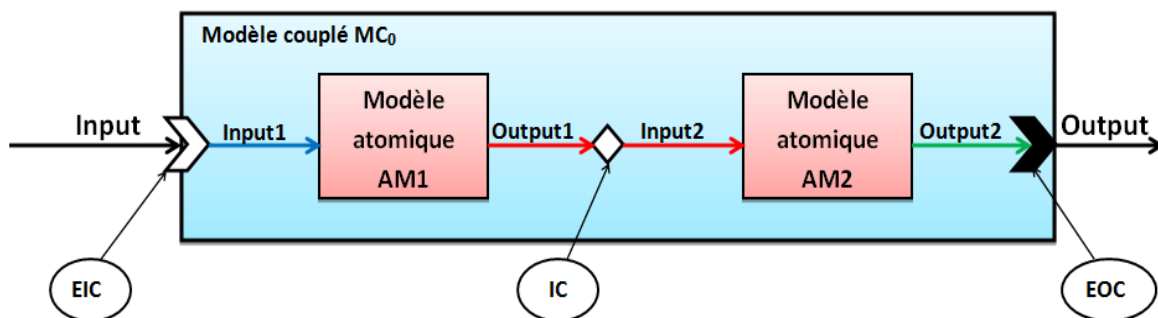


Figure II.9 : Hiérarchie de modélisation DEVS [Capoc, 2005]

1. une relation de couplage interne pour le couplage entre les ports des sous-modèles qui composent le modèle couplé (en rouge),

2. **une relation de couplage des entrées externes** pour le couplage entre les ports d'entrées du modèle couplé et les ports d'entrées des sous-modèles (en bleu),
3. **une relation de couplage des sorties externes** pour le couplage entre les ports de sorties du modèle couplé et les ports de sorties des sous-modèles (en vert).

La figure II.10 montre une hiérarchie en série entre sous-modèles d'un système. Le modèle couplé  $MC_0$  modélise au plus haut niveau ce système. Il possède un port d'entrée Input, un port de sortie Output et il contient les deux modèles atomiques  $MA_1$  et  $MA_2$ . Le modèle atomique  $MA_1$  est relié par une relation de couplage d'entrées externes au modèle couplé  $MC_0$  (en bleu), et par une relation de couplage interne au modèle atomique  $MA_2$  (en rouge sur la figure), et par une relation de couplage des sorties externes du modèle couplé  $MC_0$  (en vert sur la figure).

### C- Spécification DEVS couplé

Dans le cas du formalisme DEVS classique avec port, les spécifications d'un modèle couplé sont les suivantes :

$$MC = \langle X, Y, D, \{M_d | d \in D\}, EOC, IC, EIC \rangle \quad \text{avec :}$$

1. *l'ensemble des entrées* :  $X = \{(p,v) | p \in P_e, v \in V_x\}$  où  $P_e$  et  $V_x$  sont deux ensembles finis représentant l'ensemble des ports d'entrées et des valeurs portées par les événements reçus en entrée ;
2. *l'ensemble des sorties* :  $Y = \{(p,v) | p \in P_s, v \in V_y\}$  où  $P_s$  et  $V_y$  sont deux ensembles finis représentant l'ensemble des ports de sorties et des valeurs portées par les événements générées en sortie ;
3. *l'ensemble des modèles atomiques ou couplés* :  $D = \{m_i | i \in R^+\}$  ;
4. *modèle DEVS* :  $M_d = \langle X_d, Y_d, S_d, \delta_{ext\_d}, \delta_{int\_d}, \lambda_d, ta_d \rangle$  est un modèle atomique avec  $d \in D$ ,
5. *l'ensemble des couplages de sortie externes* :

$EOC = \{((d; b); (MC; a)) | a \in OPorts_N; b \in OPorts_d; d \in D\}$  avec  $OPorts_{MC}$  l'ensemble des ports de sortie du modèle couplé et  $OPorts_d$  l'ensemble des ports de sortie du modèle  $M_d$  ;

6. *l'ensemble des couplages internes* :

$$IC = \{((i; a); (j; b)) | i, j \in D; i \neq j; a \in OPorts_i; b \in IPorts_j\} ;$$

7. *l'ensemble des couplages d'entrées externes* :

$EIC = \{(MC; a); (d; b)) | a \in IPorts_{MC}; b \in IPorts_d; d \in D\}$  avec  $IPorts_{MC}$  l'ensemble des ports d'entrée du modèle couplé et  $IPorts_d$  l'ensemble des ports d'entrée du modèle  $M_d$ .

Brièvement, chaque modèle couplé comprend les informations suivantes :

- a) l'ensemble des modèles qui le composent,
- b) l'ensemble des ports d'entrée qui recevront les événements externes,
- c) l'ensemble des ports de sortie qui émettront les événements,
- d) les couplages entre ports d'entrée et ports de sortie des modèles composant le modèle couplé.

### II.5.2 La simulation DEVS

Comme nous l'avons dit précédemment, le formalisme DEVS repose sur une séparation explicite de la phase de modélisation et de la phase de simulation [Capoc, 2005]. Le simulateur est généré de manière automatique en se basant sur la description des modèles et de leur structure comme cela est présenté dans la Figure II.10.

Comme nous pouvons l'observer sur cette figure, à partir de la description hiérarchique des modèles, une arborescence de processus est construite ayant comme coordinateur principal la racine de l'arbre : « Root ».

Le formalisme isole deux groupes de processus. Le premier groupe représente les différents « simulateurs » associés à chaque modèle atomique. Le second groupe contient les différents « coordinateurs » associés à chaque modèle couplé. Il est important de préciser qu'un coordinateur pourra lui-même gérer des coordinateurs de niveau inférieur et être géré par des coordinateurs de niveau supérieur.

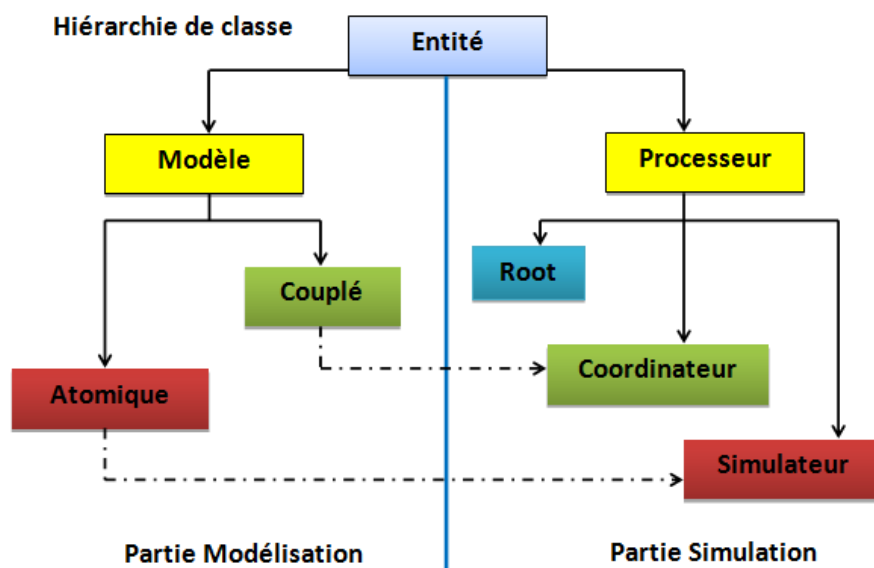


Figure II.10 : Correspondance entre modélisation et simulation DEVS [Capoc, 2005]



Le mécanisme de simulation est responsable de l'ordre d'exécution des fonctions comportementales ( $\delta_{int}, \delta_{ext}, \lambda, ta$ ) pour chaque modèle atomique DEVS. Ce mécanisme est géré par le simulateur qui, comme le coordinateur, peut recevoir ou envoyer (04) types de messages :

1. **Le message d'initialisation (i, t)** permet à tous les acteurs d'effectuer une synchronisation temporelle initiale.
2. **Le message de transition interne (\*, t)** permet la gestion d'un événement interne avec l'exécution de la fonction de transition interne  $\delta_{int}$ .
3. **Le message de sortie (y, t)** permet le transport des sorties (données par l) aux éléments parents et résulte de la réception d'un message (\*, t).
4. **Le message de transition externe (x, t)** permet la gestion d'un événement externe avec l'exécution de la fonction de transition externe  $\delta_{ext}$ .

Nous avons choisi d'illustrer les concepts de modèle atomique, modèle couplé et simulation par un exemple concret.

### II.5.3 Description d'un modèle DEVS atomique et d'un modèle DEVS couplé

Sur la Figure II.11, M1 est un modèle DEVS atomique. Sur ce modèle, les deux types de transitions existantes entre états, externe et interne, sont présentés. S1 est l'état initial.

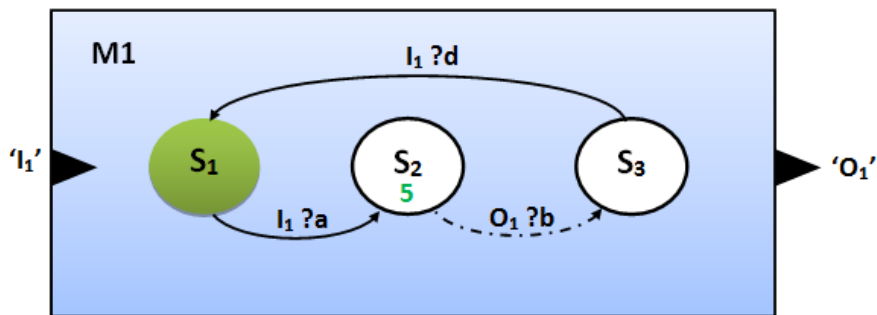


Figure II.11 : Un modèle DEVS atomique

Nous formalisons ce modèle atomique de la manière suivante :

$$M_1 = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta \rangle \quad \text{où}$$

1.  $X = \{a, d\}$ ,
2.  $S = \{S1, S2, S3\}$ ,
3.  $Y = \{b\}$ ,
4.  $\delta_{int} : S2 \rightarrow S3$ ,
5.  $\delta_{ext} : (S1, e, a) \rightarrow S2$  et  $(S3, e, d) \rightarrow S1$ ,

6.  $\lambda : S2 \rightarrow S3$ ,

7.  $ta : S2 \rightarrow 5$ .

La dynamique du modèle peut être décrite de la manière suivante : *Le modèle est dans sa phase initiale S1. Lorsque le modèle reçoit l'évènement "a" sur son port "I1", il passe dans la phase "S2". Au bout de 5 unités de temps, le modèle évolue dans la phase "S3", en émettant l'évènement "b".*

Sur la Figure II.12, M est un modèle DEVS couplé. Nous formalisons ce modèle couplé de la manière suivante :  $M = \langle X, Y, D, \{Md | d \in D\}, EIC, EOC, IC \rangle$  où

1.  $X = \{(I, a), (I, b), (I, d)\}$ ,

2.  $Y = \{(O, c)\}$ ,

3.  $D = \{M11, M12\}$ ,

4. Les modèles composants : *M11* et *M12* sont deux modèles DEVS atomiques,

5.  $EIC = \{(M, I), (M11, I_{11}), (M, I), (M12, I_{12})\}$ ,

6.  $EOC = \{(M12, O_{12}), (M, O)\}$ ,

7.  $IC = \{(M11, O_{11}), (M12, O_{12})\}$ ,

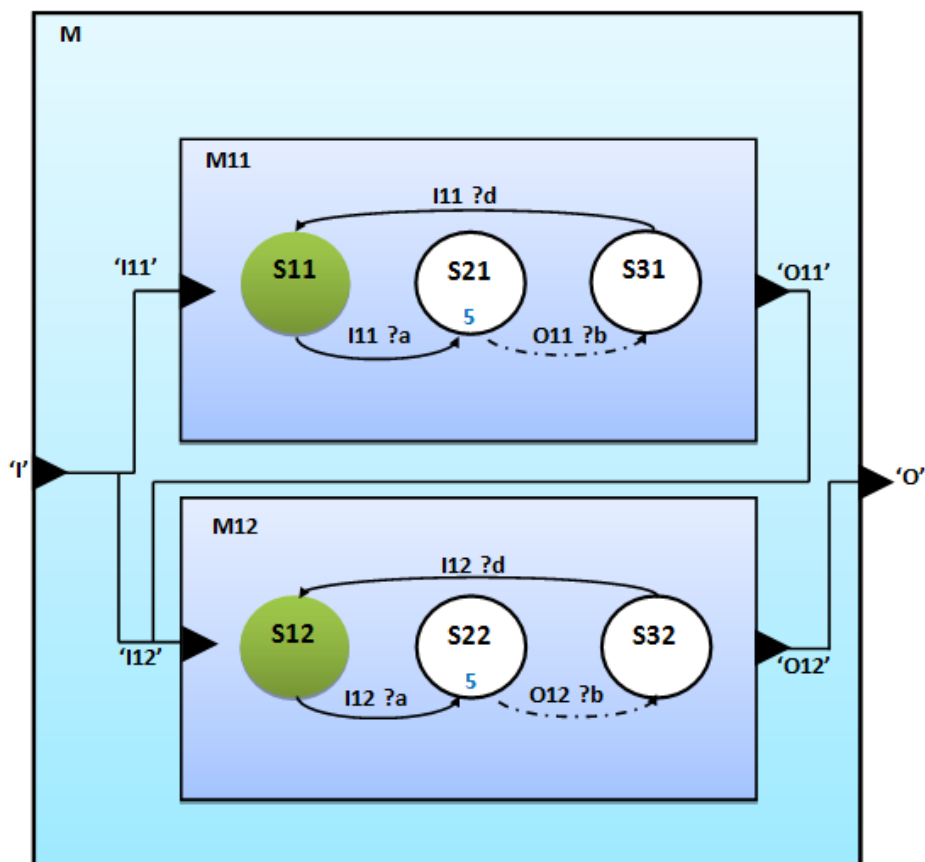


Figure II.13 : Un modèle DEVS couplé

Les deux modèles M11 et M12 sont semblables au modèle atomique décrit ci-dessus. M11 et M12 sont initialement dans leur phase respective S11 et S12. Lorsque le modèle couplé M reçoit un événement sur son port "I", il l'envoie à M11 et à M12, sur les ports respectifs I<sub>11</sub> et I<sub>12</sub>. Si l'événement reçu par "I" est "a", M11 évolue dans la phase S21, et M12 ne change pas de phase. Au bout de 5 unités de temps, M11 émet l'événement "b" par son port de sortie O<sub>11</sub> et le transmet à M12 sur le port I<sub>12</sub>. Si M12 est dans sa phase initiale S12, il évolue dans S22, et émet l'événement "c" par son port de sortie "O<sub>12</sub>", sur le port de sortie "O" du M. Si l'événement reçu par "I" est "b", M12 évolue dans la phase S22, et M11 ne change pas de phase. Au bout de 5 unités de temps, M12 émet l'événement "c" par son port de sortie O<sub>12</sub> et le transmet à M sur le port O.

## **II.6 Conclusion**

Ce chapitre a permis de définir les concepts généraux de la modélisation et de la simulation. Parmi la multitude d'approches et de systèmes pouvant être modélisés nous nous sommes intéressés en particulier aux systèmes à évènements discrets qui sont de nos jours très largement utilisés. Pour cela, nous avons référencé les principales méthodes et approches associées (automates à états finis, réseaux de Pétri, systèmes multi-agents, DEVS) afin d'en choisir une.

En adéquation avec les finalités recherchées, à savoir l'optimisation par la simulation, nous avons choisi de baser nos travaux sur un formalisme de haut niveau, robuste et extensible : le formalisme DEVS. Nous avons alors effectué une analyse critique des rares travaux d'optimisation via simulation impliquant celui-ci. Ces différentes approches, qu'elles soient génériques ou spécifiques ont permis l'ouverture d'une voie de recherche plus que prometteuse.

Cependant de nombreuses pistes restent à explorer tant au niveau du dynamisme, de la généralité que des performances. Les problématiques qui en découlent ont servi de socle à notre architecture de modèle.

Celle-ci devra permettre : le dynamisme de l'optimisation, l'arrêt anticipé de certaines simulations, une adaptation « sur mesure » automatique au problème étudié, la réutilisation de l'ensemble des composants et ce de manière efficace et intelligente. Les concepts développés ainsi que leur formalisation DEVS sont présentés dans le chapitre suivant.

# **CHAPITRE III**

Architectures des systèmes &  
modélisation DEVS.

## **Chapitre III : Architectures des systèmes et modélisation DEVS.**

### **III.1 Introduction**

Dans ce chapitre nous allons modéliser des objets connectés à haut niveau d'abstraction après avoir défini un environnement permettant la simulation de capteurs sans fil dédiés à un bâtiment intelligent. Dans ce travail, une démarche basée sur la localisation de l'utilisateur est envisagée pour concevoir, via la modélisation DEVS, un système de gestion de l'énergie utilisé pour planifier la consommation d'énergie des appareils intelligents installés dans les bâtiments intelligents pour rationaliser leur consommation d'énergie.

Afin de rationaliser et de simplifier ce travail, nous avons adopté le plan suivant :

1) Nous avons adopté un scénario pour appliquer le contrôle de la consommation d'énergie au sein d'un bâtiment intelligent. C'est un bâtiment connecté, contenant des équipements consommateurs de l'énergie, qui ont le rôle de fournir le confort à un utilisateur accompagné toujours avec son smartphone.

2) Suite au déroulement de ce scénario, une architecture système a été conçue comme un modèle afin de réaliser les stratégies et les scénarios inclus dans cet environnement selon l'architecture multi-niveaux suggéré dans [Maato, 2014]. Dans ce travail, nous y avons ajouté de nouveaux composants, qui sont en ligne avec l'ajustement de la consommation d'énergie suivant la localisation de l'utilisateur.

3) Nous avons modélisé selon le formalisme DEVS, tout ce qui existerait dans l'environnement qui correspond à la scène susmentionnée.

## **III.2 L'environnement d'application**

Comme nous l'avons dit, l'application effectuée dans l'environnement d'application est facilitée par des capteurs sans fil déployés dans l'atmosphère du bâtiment pour mesurer la température, l'humidité ou différents niveaux de gaz. Dans notre cas, WSN-BAS assure l'échange d'informations entre tous les objets connectés, y compris les capteurs du sol, et leurs lectures peuvent être combinées pour former une mesure fiable. Les capteurs et les actionneurs se coordonneront avec les serveurs de brouillard internes en utilisant des décisions et des actions distribuées pour interagir rapidement avec les données. WSN-BAS maximise son utilité en ajustant ses objectifs opérationnels en fonction de l'état du système et de l'emplacement de l'utilisateur.

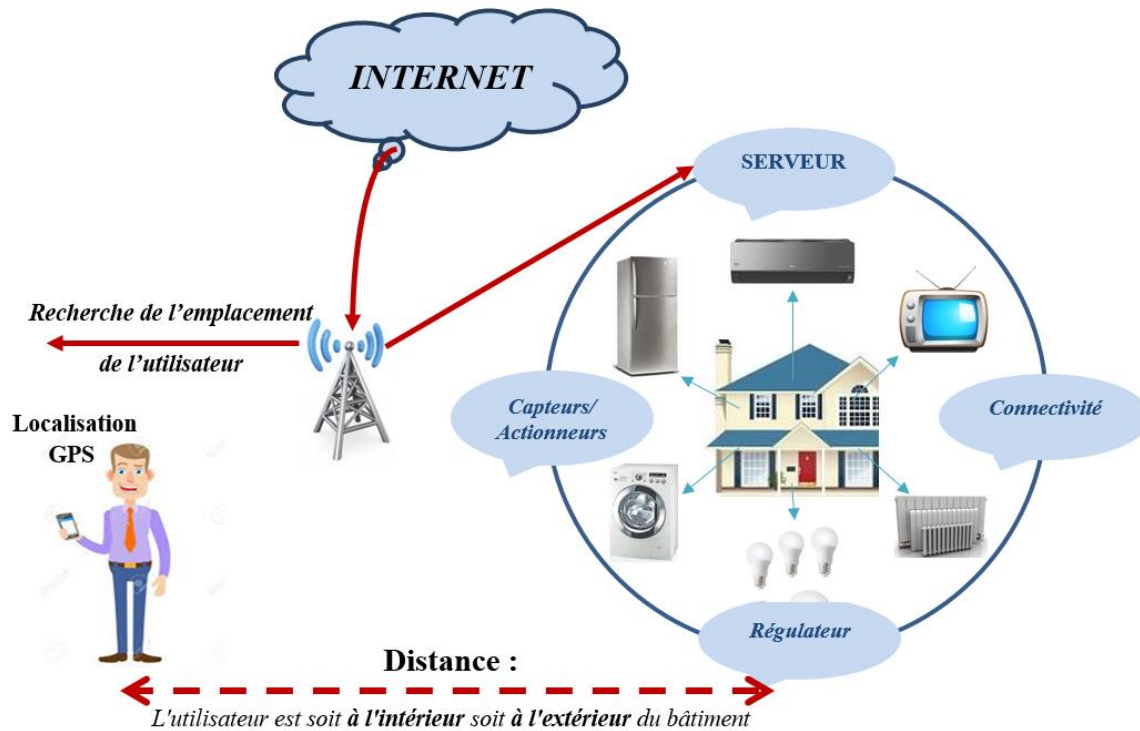
Deux algorithmes de planification sont conçus pour optimiser la consommation d'énergie des bâtiments en déterminant la localisation des utilisateurs du bâtiment, à savoir l'algorithme de localisation des utilisateurs et l'algorithme d'ajustement des économies d'énergie. Chaque algorithme sert une tâche distincte et WSN-BAS passe d'une stratégie de planification à une autre en fonction de la décision du serveur Fog. Comme nous l'avons dit, avec l'application du calcul du brouillard dans ce scénario, les bâtiments intelligents peuvent maintenir leur structure interne et leurs environnements pour économiser l'énergie et même d'autres ressources [Stojm, 2014].

### **III.2.1 Description de la scène générale**

La scène que nous présentons maintenant résume tous les scénarios saillants de notre environnement. Notre environnement est constitué d'un occupant d'un bâtiment qui peut le quitter pour aller ailleurs. Cet utilisateur est toujours accompagné de son smartphone qui dispose d'un service de localisation GPS. Nous voulons contrôler automatiquement le mode de fonctionnement des appareils électroménagers dans tous les cas d'utilisation, et nous voulons également effectuer ce contrôle localement (Figure III.1).

L'idée de base est que dès que l'utilisateur quitte le bâtiment, le système de gestion technique du bâtiment doit immédiatement éteindre les équipements consommateurs d'énergie, et si cela n'est pas possible, il réduira au moins la consommation d'énergie. L'objectif est de créer un mécanisme permettant de rationaliser automatiquement l'utilisation de l'énergie dans le bâtiment.

Pour cela, nous suivrons trois politiques de consommation d'énergie dans notre bâtiment, une lorsque l'utilisateur est à l'intérieur du bâtiment, une lorsqu'il est à l'extérieur et une lorsque sa



**Figure III.1 : Politique de localisation de notre serveur local**

position est inconnue. L'idée est de toujours localiser son Smartphone à pleine vitesse. Son support de Smartphone a quitté le bâtiment et s'est installé à l'extérieur. Le changement d'emplacement déclenche le serveur pour ajuster automatiquement les stratégies d'économie d'énergie pour le bâtiment. La fonction principale est de permettre au serveur de détecter les changements d'emplacement de l'utilisateur et d'apporter des modifications à la politique énergétique en allumant/éteignant les appareils dans ce bâtiment et en élevant/abaissant la consommation de certains appareils.

Une authentification appropriée garantit que notre système est autonome, séparé et isolé tout au long de son fonctionnement, et n'autorise pas deux connexions avant de se déconnecter [Cesco, 2020] Par conséquent, le premier algorithme sera, après l'authentification correcte du système, localiser l'utilisateur et le deuxième algorithme sera pour modifier la politique d'économie d'énergie de la maison en fonction de l'emplacement de l'utilisateur.

### III.2.2 Les algorithmes généraux

Le premier algorithme a l'avantage que si l'autorisation de l'utilisateur échoue, la session de l'utilisateur est fermée. Pendant ce temps, l'accès de l'utilisateur est bloqué et l'utilisateur est averti en conséquence. L'authentification de l'utilisateur va être réalisée sur la base du nom

d'utilisateur et du mot de passe saisis. Il existe d'autres types d'algorithmes permettant aux utilisateurs de s'authentifier efficacement lorsqu'ils sont dans des emplacements spécifiques dans un environnement et seront rejetés s'ils changent d'emplacement [Abyan, 2019] [Belhi, 2020], mais dans ce travail nous sommes satisfaits d'une authentification ordinaire.

---

**Algorithme 1** : Localisation de l'Utilisateur (LU)

---

```

                                                                    /* initialisation */
Entrées : Accès_Autorisé, Authentification
Sortie : SL, LU
1 Tantque Accès_Autorisé = faux faire
2   Si Authentification = vrai alors
3     | Accès_Autorisé ← vrai;
4   sinon
5     | Accès_Autorisé ← faux;
6   Si Accès_Autorisé = vrai alors                                /* requête SL */
7     | si SL n'est pas disponible alors                            /* requête Cloud */
8     | | obtenir SL via les centres de données Cloud ;
9     | sinon                                                        /* requête Serveur */
10    | | obtenir SL via le serveur;
11  Si SL est disponible alors
12  | retournez LU ;                                             /* renvoie de la localisation */
```

---

Donc pour la localisation de l'utilisateur, le processus n'a lieu qu'après une authentification correcte pour accéder aux services de localisation (voir Algorithme 1). Ces services peuvent être obtenus auprès du serveur et s'ils ne sont pas disponibles, ils seront obtenus depuis le Cloud. Après avoir accédé à l'utilisateur lieu, il est présenté à la partie du système qui demande son utilisation dans ses procédures. Passant maintenant au deuxième algorithme : l'ajustement automatique de la politique de consommation d'énergie sur la base de la localisation de l'utilisateur (Voir algorithme 2). Nous considérons cet algorithme comme la pierre angulaire du processus d'optimisation attendu de nos conceptions proposées, qui contrôlera automatiquement le mode de consommation d'énergie du bâtiment intelligent en fournissant un plan de consommation d'énergie, qui sera exécuté par des actionneurs intégrés dans les installations d'appareils électroménagers.



**Algorithme 2 :** Plan\_de\_consommation\_d'énergie

**Entrées :**  $LU, D, Min_{puiss\_exist}$

**Sortie:**  $Plan_{Bâtiment}$

```

1  Tantque  $LU$  est disponible faire
2      Calculer  $Bâtiment_{puiss\_cons}$ ;
3      Calculer  $D_{Bâtiment-User}$ ;
4      Régler  $Mode\_Bâtiment$ ;
5      Si ( $(D_{Bâtiment-User} \leq D)$  et ( $Bâtiment_{puiss\_cons} \geq Min_{puiss\_exist}$ )) alors
           /* l'utilisateur est dans le bâtiment*/
6           $Mode\_Bâtiment \leftarrow Luxe$ ;
7           $Plan_{Bâtiment} = 1$ ;
8      sinon /* l'utilisateur est à l'extérieur */
9           $Mode\_Bâtiment \leftarrow Econome$ ;
10          $Plan_{Bâtiment} = 0$ ;
11 Retourner  $Plan_{Bâtiment}$ ;

```

En effet, cet algorithme modifie la mode de consommation d'énergie dans le bâtiment en fonction des besoins des utilisateurs, de la quantité d'énergie disponible et des prévisions météorologiques. Ceci est fait après avoir calculé la distance entre l'utilisateur et le bâtiment.

Après avoir ajusté les modes de consommation d'énergie en fonction du climat, si la distance entre l'utilisateur et le bâtiment est inférieure à une distance spécifique au périmètre du bâtiment que nous supposons être de 5 mètres, et que l'énergie est disponible, cet algorithme modifie directement le mode de l'énergie dans le bâtiment au mode luxe, mais si l'utilisateur est à l'extérieur, le bâtiment conservera son mode économique.

En supposant que l'utilisateur ne peut pas occuper le bâtiment plus de 8 heures. Cet algorithme ajuste automatiquement le mode de consommation d'énergie en éteignant les appareils temporaires et en minimisant la consommation du reste des appareils permanents pendant 16 heures.

De cette façon, nous réduisons la consommation énergétique des bâtiments au moment où les utilisateurs quittent leurs bâtiments, c'est-à-dire sans compromettre leurs désirs de luxe. On peut donc dire qu'avec ce mécanisme automatisé, on peut garantir un maximum de bien-être aux utilisateurs des bâtiments intelligents au moindre coût.

### III.3 Contrôle du bâtiment basant sur la localisation de l'utilisateur

Le déploiement de capteurs sans fil pour mesurer la température, l'humidité ou divers niveaux de gaz dans l'environnement du bâtiment facilite les applications dans ce scénario. Dans ce cas, des informations peuvent être échangées entre tous les capteurs au sol et leurs lectures peuvent être combinées pour former des mesures fiables. Les capteurs réagiront aux données en utilisant des décisions et des activations distribuées sur les dispositifs de brouillard.

Les composants du système peuvent alors travailler ensemble pour abaisser la température, injecter de l'air frais ou ouvrir les fenêtres. Les climatiseurs peuvent éliminer l'humidité de l'air ou augmenter l'humidité. Les capteurs peuvent également suivre et réagir au mouvement (comme allumer ou éteindre les lumières). Les appareils Fog peuvent collaborer à un niveau de conduite plus élevé. Le calcul du brouillard est appliqué dans cette scène., les bâtiments intelligents peuvent maintenir leur structure interne et leur environnement pour économiser de l'énergie, de l'eau et même d'autres ressources [Stojm, 2014].

#### III.3.1 Comment localiser l'utilisateur ?

Nous présentons maintenant un scénario qui résume toutes les scènes saillantes de notre environnement en contrôlant automatiquement le fonctionnement des appareils électroménagers dans tous les cas d'utilisation.

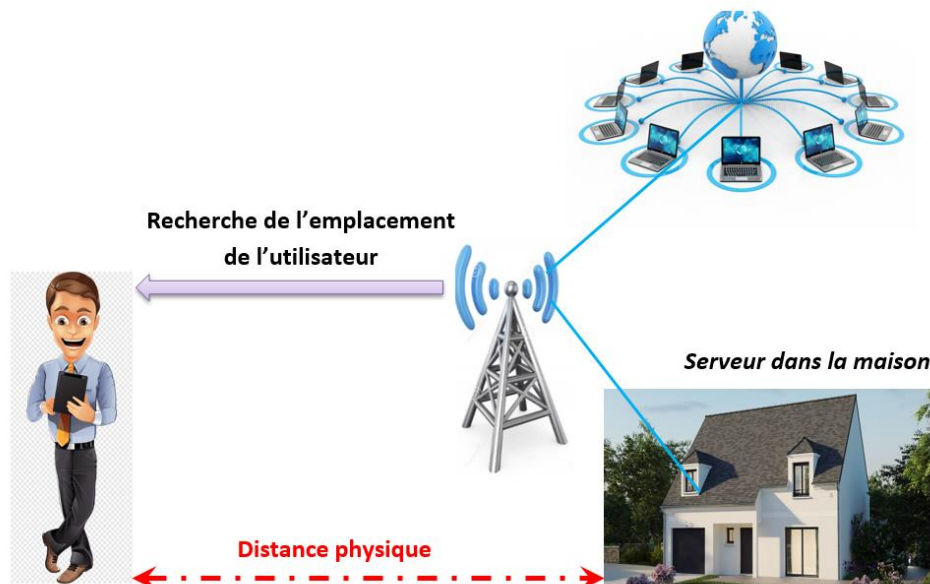


Figure III.2 : Comment localiser l'utilisateur dans notre scénario ?

Le mécanisme consiste à placer des capteurs à l'intérieur du domicile de l'utilisateur, connectés au réseau local et installés à côté de tous les appareils électroménagers pour mesurer l'énergie consommée et le luxe économisé, et à installer un serveur lié au réseau à l'intérieur du bâtiment.

En obtenant les informations de localisation de l'utilisateur à l'aide de services de positionnement GPS, envoyés à l'avance aux sites de déploiement du smartphone de l'utilisateur, le serveur installé à l'intérieur du bâtiment peut rapidement télécharger le contenu traduit [Luan, 2015]. Le but de cette méthode est qu'une fois le l'utilisateur quitte le bâtiment, le système de gestion du bâtiment entre en mode économe.

Nous suggérons que lorsque l'utilisateur quitte son immeuble et avec son Smartphone, les changements d'emplacement déclenchent le serveur pour ajuster automatiquement la politique d'économie d'énergie en allumant ou en éteignant les appareils ou en économisant et en augmentant l'énergie dans ce bâtiment.

### **III.3.2 L'ajustement automatique de la politique d'économie d'énergie**

Nous avons présenté un organigramme pour le ciblage des utilisateurs et l'ajustement automatique de la politique d'économie d'énergie que nous avons expliqué précédemment. C'est ce qu'illustre la Figure III.3. Il montre étape par étape comment notre travail est effectué du début à la fin. Il est divisé en deux parties : le positionnement de l'utilisateur et l'ajustement de la stratégie d'économie d'énergie qui en résulte.

Le travail commence après une authentification appropriée à l'application de réglage de la politique d'économie d'énergie et se termine lorsque l'utilisateur souhaite quitter l'application, c'est-à-dire qu'une fois que l'utilisateur ouvre l'application, le réglage automatique commence à fonctionner et une fois l'application fermée, le réglage automatique cesse de fonctionner ; cependant, si le bâtiment n'est pas économe en énergie, l'utilisateur ne pourra pas quitter l'application.

De plus, nous supposons que l'utilisateur occupe la maison pendant une période n'excédant pas 8heures (8 heures à la maison pour le repos, 16 heures entre leur travail, et ses besoins). En conséquence, l'ajustement automatique des politiques de consommation d'énergie arrêtera le fonctionnement de certains appareils électriques pendant 16 heures dans la maison.

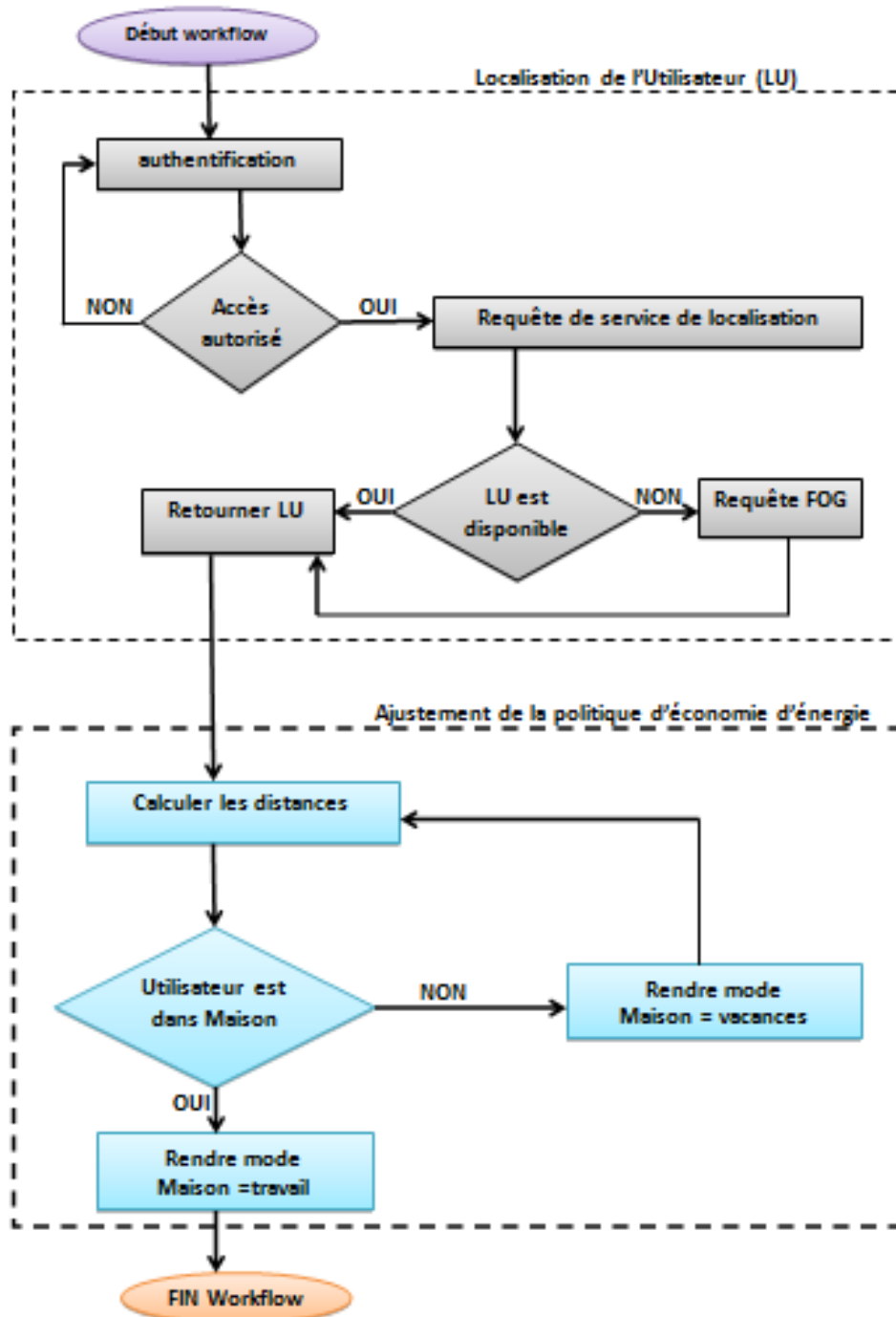


Figure III.2 : Flux de travail de l'économie d'énergie

### III.4 Conception du système et modélisation DEVS

Compte tenu de l'application de ce scénario à un environnement constitué d'un utilisateur constamment accompagné de son smartphone, occupant un bâtiment équipé d'équipements énergivores pour apporter le confort souhaité, la conception initiale des systèmes résultants sera

composé de sous-systèmes principaux, chacun rassemble un ensemble d'objets et correspond à un acteur principal du scénario (Bâtiment, Occupant, Serveur, Connectivité, Capteurs/Actionneur, Régulateur et Equipements).

### III.4.1 L'architecture des systèmes résultants

Le bâtiment représentera l'ensemble du système et l'utilisateur sera représenté par son téléphone portable (l'objet connecté au système est son smartphone). Nous considérons le serveur dans notre système comme un acteur principal de l'optimisation de la consommation d'énergie dans le

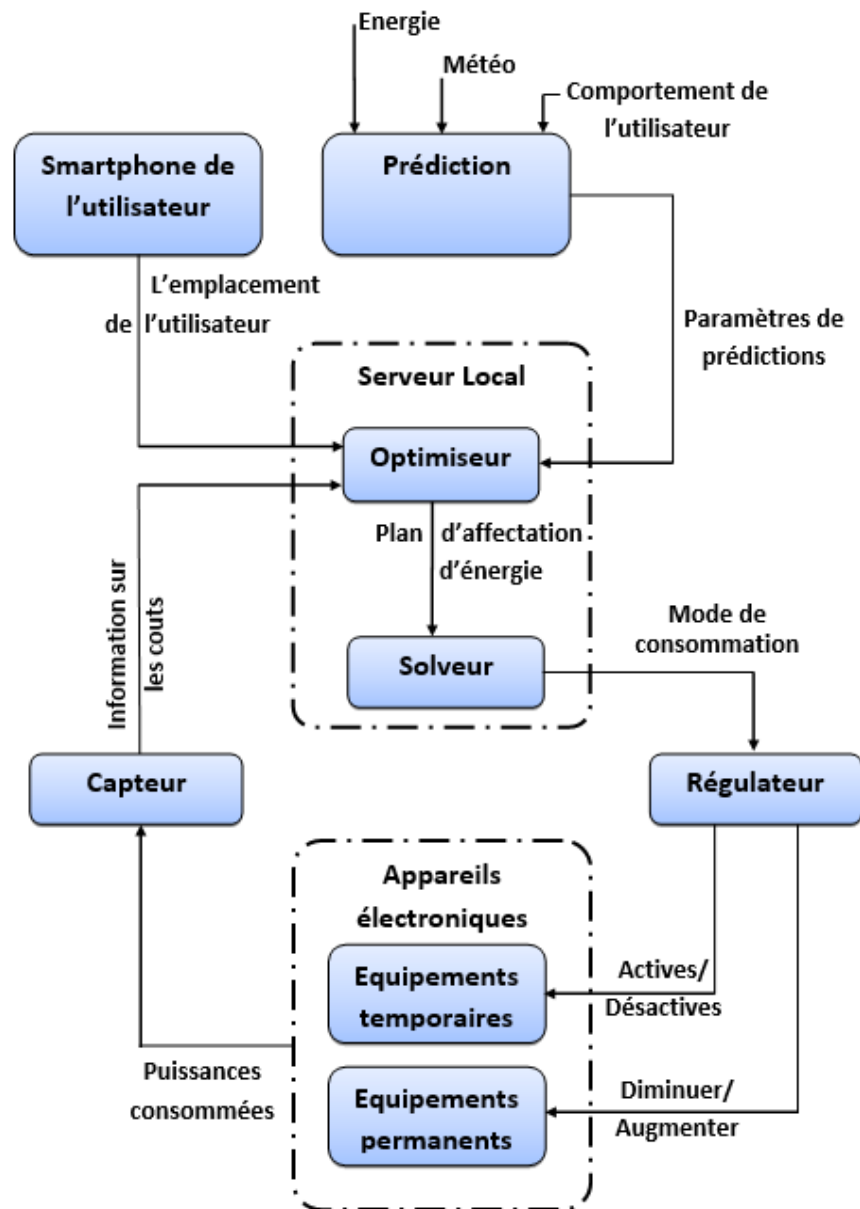


Figure III.4 : Système d'optimisation de la consommation d'énergie dans bâtiments

bâtiment ; si la localisation de l'utilisateur change par rapport au bâtiment (intérieur bâtiment/extérieur bâtiment), c'est lui qui va trouver une solution rapide pour l'ajustement automatique de la politique d'utilisation de l'énergie dans le bâtiment. Ce serveur couvre deux rôles principaux dans le système, le rôle de l'optimiseur et du solveur. Sur la base des données d'anticipation (prédictions), y compris le niveau d'énergie, la météo et le comportement de l'utilisateur, sur la base de ce qui a été détecté en terme d'énergie consommée et le coût résultant (l'énergie disponible répondra-t-elle à la situation requise ?), et sur la base de l'emplacement de l'utilisateur, l'optimiseur va préparer et présenter un plan général de consommation énergétique du bâtiment au solveur.

Ce dernier résout l'équation et la solution sera le mode de consommation d'énergie à adopter. Les équipements seront soumis à l'application de ce mode, déjà décidé, sous le commandement de ce qu'on appelle le régulateur. Ce dernier contrôle en fait la consommation d'énergie des équipements (les équipements temporaires avec les instructions activer/désactiver, et les équipements permanents avec les instructions augmenter/diminuer).

Les capteurs surveillent et collectent les informations de consommation d'énergie de tous les équipements en temps réel et en continu. Alors que les actionneurs, nous les considérons comme contenus dans les appareils électroménagers eux-mêmes (les appareils électroménagers deviennent intelligents et connectés). La figure III.4 au-dessus représente le système que nous avons proposé pour contrôler automatiquement la consommation d'énergie au sein d'un bâtiment intelligent, et elle comprend tout ce qui vient d'être expliqué.

### **III.4.2 Modélisation DEVS**

Dans ce travail, nous utilisons la notation graphique de [Zeigl., 2000] comme le montre la figure 4, afin d'illustrer les aspects comportementaux de tous les modèles atomiques qui entrent dans la composition de notre système résultant. Un état est représenté par un cercle contenant le nom, les opérations sur les variables et la durée de vie de l'état ( $t_a$ ). Le modèle est dans son état initial "Si". La figure III.5 (a) représente une transition externe. Un événement d'entrée est spécifié à l'aide de "?". La ligne continue représente une transition d'état spécifiée par la fonction de transition externe. Lorsque le modèle reçoit l'événement "m" sur son port "in", il passe à l'état "Sj". La figure III.5 (b) désigne une transition interne. Un événement de sortie est spécifié à l'aide de "!". La ligne pointillée représente une transition d'état spécifiée par la fonction de transition interne. Le modèle évolue dans l'état "Sj", émettant l'événement "m".

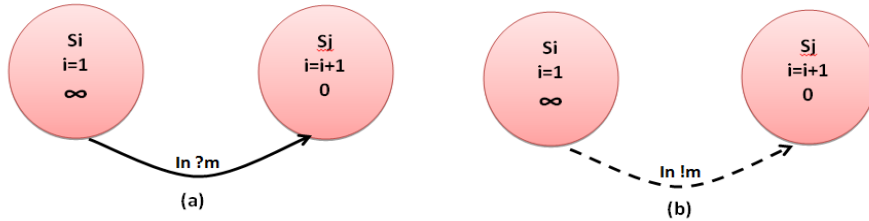


Figure III.5 : Notation graphique d'une transition externe (a) et interne (b)

Notons ici que le travail présenté dans [Aid, 2016], explique clairement ce genre de notation graphique aux modèles atomiques, auxquels ressemblent nos modèles mais dont le contenu est assez différent.

Maintenant, nous présentons les modèles couplés décrivant la composition des objets qui contribuent à l'optimisation de la consommation d'énergie dans le bâtiment. Dans notre environnement proposé, nous notons qu'ils sont identiques dans la maison. De plus, nous devons noter que l'environnement DEVS est un outil modulaire, car l'utilisateur peut étendre n'importe quel nombre de composants présentés ici, tels que les bâtiments ou leur contenu [Maato, 2014] Par conséquent, l'utilisateur peut facilement ajouter et/ou supprimer des composants pour modéliser et simuler les différentes structures.

Le modèle que nous proposons divise chaque SGEB (EMS Maison), en sept sous-systèmes : Smartphone de l'utilisateur, Prédiction, Optimiseur, Solveur, Régulateur, Capteurs et Equipements [Maato, 2021].

La figure III.6 présente le diagramme de la structure du système de gestion d'énergie dans un bâtiment en DEVS.

Nous donnons à SGEB la structure suivante d'un modèle DEVS couplé :

$$SBEB = (X, Y, D, \{M_d / d \in D\}, EIC, EOC, IC) \quad \text{où :}$$

- $X = \{(p, v) / p \in IPorts, v \in X_p\}$  sont les valeurs d'entrées ;
- $Y = \{(p, v) / p \in OPorts, v \in Y_p\}$  sont les valeurs de sortie ;
- $D$  = l'ensemble des noms des modèles atomiques ou couplés qui composent le SGEB :  
 -  $D = \{Smartphone\ de\ l'utilisateur, Prédiction, Optimiseur, Solveur, Régulateur, Capteurs, Equipements\}$  ;
- $M_D$  est l'ensemble des modèles atomiques ou couplés qui composent le SGEB :

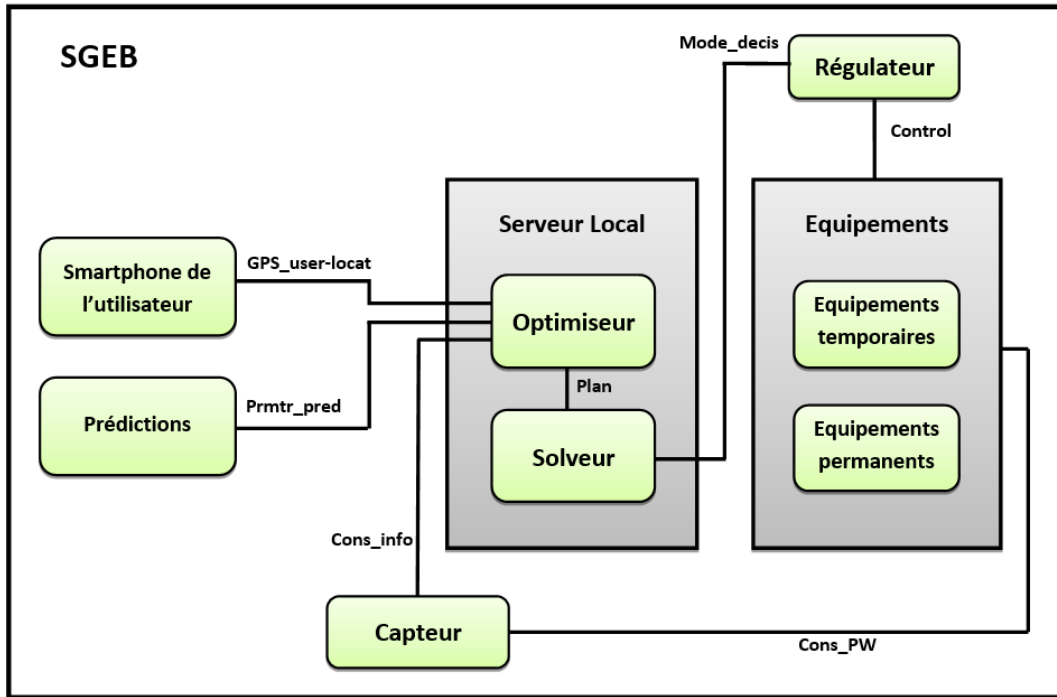


Figure III.6 : Système de gestion de l'Energie dans un Bâtiment (SGEB) en DEVS

- $M_{Capteurs}$  est le modèle du sous-système *Capteurs* ;
- $M_{Prédictions}$  est le modèle du sous-système *Prédictions* ;
- $M_{Optimiseur}$  est le modèle du sous-système *Optimiseur*, et ainsi de suite ;
- $EIC = \{((BEMS, a), (d, b)) / a \in IPorts_s, b \in IPorts_d\}$
- $EOC = \{((d, b), (BEMS, a)) / a \in OPorts_{BEMS}, b \in OPorts_d\}$
- $IC = \{((i, a), (j, b)) / i, j \in D, i \neq j, a \in OPorts_i, b \in IPorts_j\}$

EIC définit les ports du modèle EMS qui sont connectés aux ports des modèles de composants qui reçoivent des événements externes ; EOC définit les ports du modèle EMS qui sont connectés aux ports des modèles de composants qui émettent des événements et IC sont le couplage interne entre les modèles composants.

En raison du grand nombre de modèles atomiques dans le modèle globale et des similitudes dans la manière dont ils sont décrits, dans ce mémoire, nous suffirons à décrire les modèles les plus importants du système. Plus clairement, pour la couche locale, nous décrivons le modèle des équipements temporaires, et le modèle des équipements permanents ; Pour la couche reactive, le modèle de solveur et le modèle de régulateur ; Et pour la couche anticipative, le modèle d'optimiseur.



En ce qui concerne Smartphone de l'utilisateur, ce qui nous intéresse dans cette partie de modélisation, ce sont les informations de localisation de l'utilisateur que ce dernier donne à l'optimiseur à tout moment. Sur la base de ces informations, l'emplacement de l'utilisateur par rapport à la maison, peut être facilement déterminé.

### III.4.2.1 Modélisation des équipements temporaires

Les équipements temporaires sont des appareils électroménagers qui sont activés pendant une période de temps spécifique et limitée pour être désactivés une fois cette période expirée. Ce sont des éléments de la couche locale. Le but de la présence d'un équipement temporaire dans le bâtiment est de fournir un service temporaire à l'utilisateur. Il ne peut être qu'activé ou désactivé. La figure III.7 présente le modèle atomique DEVS d'un équipement temporaire avec quatre états : Off, Essai\_On, On et Essai\_Off.

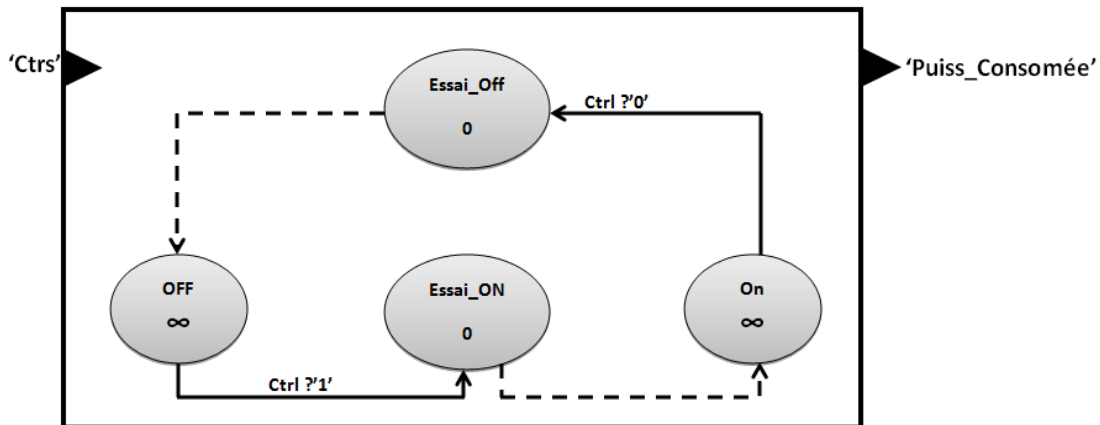


Figure III.7 : Modèle atomique DEVS d'une lampe simple

La description formelle DEVS du modèle « Equipt\_Tempor » est :

$$Equipt\_Tempor = (X, S, Y, \delta_{ext}, \delta_{int}, \lambda, ta) \quad \text{où:}$$

- $X = \{0, 1\}$ , représente les valeurs d'entrée;
- $Y = \{ON, OFF\}$ , représente les valeurs de sortie;
- L'ensemble des variables d'état:  $S = \{(phase, ta)\}$ , où:
  - $Phase = \{On, Essai\_On, Essai\_Off, Off\}$ ;
  - $\delta_{ext}(OFF, In?'1') = (Essai\_On, 0)$ , l'événement extérieur  $in?'1'$  mettre l'équipement à l'état de marche;  $\delta_{ext}(ON, In?'0') = (Essai\_Off, 0)$ , l'événement extérieur  $in?'0'$  mettre l'équipement à l'état éteint;

- $\delta_{int}(Essai\_On) = (On), \delta_{int}(Essai\_Off) = (Off);$
- $\lambda(Essai\_On) = ON, \lambda(Essai\_Off) = OFF;$
- $ta(On) = \infty, ta(Off) = \infty, ta(Essai\_On) = 0, ta(Essai\_Off) = 0.$

### III.4.2.2 Modélisation des équipements permanents

Passant maintenant aux appareils persistants, la consommation d'énergie de ces appareils peut être ajustée. La consommation d'énergie est généralement définie en pourcentage. La Figure III.8 représente le modèle atomique DEVS d'un dispositif permanent. Ce mode peut être changé pour les états On, Off, Test\_On, Test\_Off, Standby, Raise et Lower. En d'autres termes, le modèle doit être dans un seul de ces différents états à un moment donné. Par conséquent, la consommation peut être réduite uniformément de 0 à 100 % ou de 100 à 0 %.

La description formelle DEVS du modèle "Equipt\_permanent" est :

$$Equipt\_Permanent = (X, S, Y, \delta_{ext}, \delta_{int}, \lambda, ta) \quad \text{Où:}$$

- $X = \{CCE, VEC, COO\}$ , présente les variables d'événement d'entrée, où:

–CCE : Control de la Consommation d'Énergie c à d augmenter ou diminuer la consommation d'énergie.  $CCE = \{0, 1\}$ , l'événement externe  $CCE = '0'$  indique une diminution de la consommation; l'événement externe  $CCE = '1'$  indique une augmentation de la consommation.

– VEC : Valeur de l'Énergie Consommée, c'est la valeur absolue de la réduction.  $VEC \in [0, 100]$ .

– COO : Commande ON / OFF :

$COO = '0'$  indique 'Off' ;  $COO = '1'$  indique 'On'.

- $Y = \{ARRET, BAS, MOYEN, HAUT, MAX\}$ ,

- L'ensemble des variables d'état :  $S = \{(phase, Val, ta)\}$ , Où :

–Phase = {On, Off, Essai\_On, Essai\_Off, Attente, Augmenter, Diminuer}

–Val  $\in [0, 100]$  est une variable qui représente le niveau de consommation actuel.

- $\delta_{ext}(Off, COO?'1') = (Essai\_On, Val = 0, 0), \delta_{ext}(On, COO?'0') = (Essai\_Off, Val = 0, 0), \delta_{ext}(Attente, CCE?'1') = (Augmenter, Val = Val+VEC, 0), \delta_{ext}(Attente, CCE?'0') = (Diminuer, Val = Val-VEC, 0), \delta_{ext}(Attente, COO?'0') = (Essai\_Off, Val = 0, 0);$
- $\delta_{int}(On) = (Attente), \delta_{int}(Essai\_On) = On, \delta_{int}(Diminuer) = Attente, \delta_{int}(Essai\_Off) = Off, \delta_{int}(Augmenter) = Attente;$

- $\lambda(On) = on$ ,  $\lambda(Augmenter) = Haut$ ,  $\lambda(Diminuer) = Bas$ ,  $\lambda(Essai\_On) = Moyen$ ,  $\lambda(Essai\_Off) = Moyen$  ;
- $ta(Off)=\infty$ ,  $ta(On)=\infty$ ,  $ta(Essai\_On)=0$ ,  $ta(Essai\_Off)=0$ ,  $ta(Attente)=\infty$ ,  $ta(Augmenter)=0$ ,  $ta(Diminuer)=0$ .

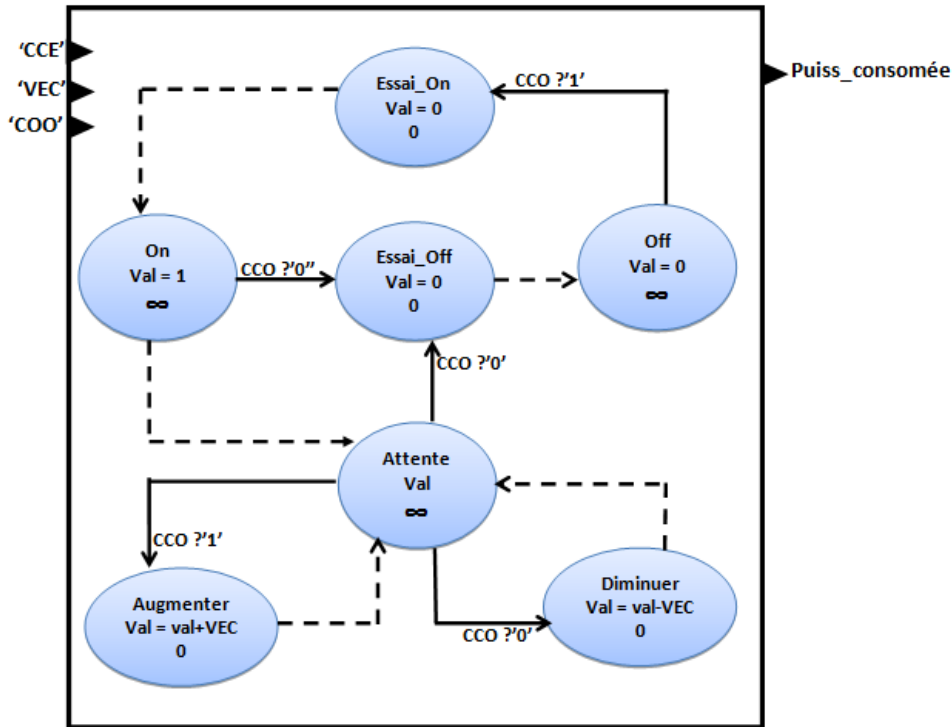


Figure III.8 : Modèle atomique DEVS d'un équipement permanent

L'état initial «Off» est passif (c'est-à-dire que sa durée de vie est  $\infty$ ). Une transition se produit au niveau de l'événement externe COO? '1' (correspondant à la commande On); ce qui nous amène à l'état «Essai\_On». La transition interne suivante permet au modèle de passer à l'état «On». L'état d'arrivée «On» de cette transition est aussi passif : son rôle consiste à mettre à jour la valeur de la variable 'Val' à 1 (la valeur minimale de consommation étant de 1%). La transition interne suivante atteint un nouvel état passif, appelé «Attente», qui protège la valeur de la variable 'Val'. Dans cet état, le modèle attend de recevoir un signal externe.

Lors d'un nouveau signal, CCE?'1', on atteint l'état transitoire «Augmenter»: son rôle consiste à augmenter la consommation (c'est-à-dire à changer la valeur de la variable Val en Val+VEC, où VEC est la valeur de variation). La transition interne suivante permet de retourner à l'état 'Attente'. Lorsque l'événement CCE? '0' survient, un nouvel état transitoire est atteint, «Diminuer», son rôle consiste à diminuer la valeur de la consommation (c'est-à-

dire changer la valeur de la variable Val en Val- VEC). La transition interne suivante permet de revenir à l'état 'Attente'. A partir de cet état, nous atteignons l'état «Essai\_Off» lors de l'événement externe CO?'0' (correspondant à la commande Off). La transition interne suivante permet au modèle de revenir à son état initial. La fonction de sortie génère une sortie externe, « Puiss\_consmée », juste après qu'aucune transition interne n'ait lieu.

### III.4.2.3 Modélisation de l'optimiseur

Passons maintenant à la couche des attentes, où l'optimiseur fournit un plan pour optimiser la consommation d'énergie en mettant en œuvre un processus qui représente un modèle mathématique pour résoudre le problème général de l'emplacement de l'utilisateur. Dans un autre sens, l'optimiseur fournit le plan d'allocation des ressources énergétiques à la couche de réaction pour une exécution réelle basée sur l'emplacement de l'utilisateur.

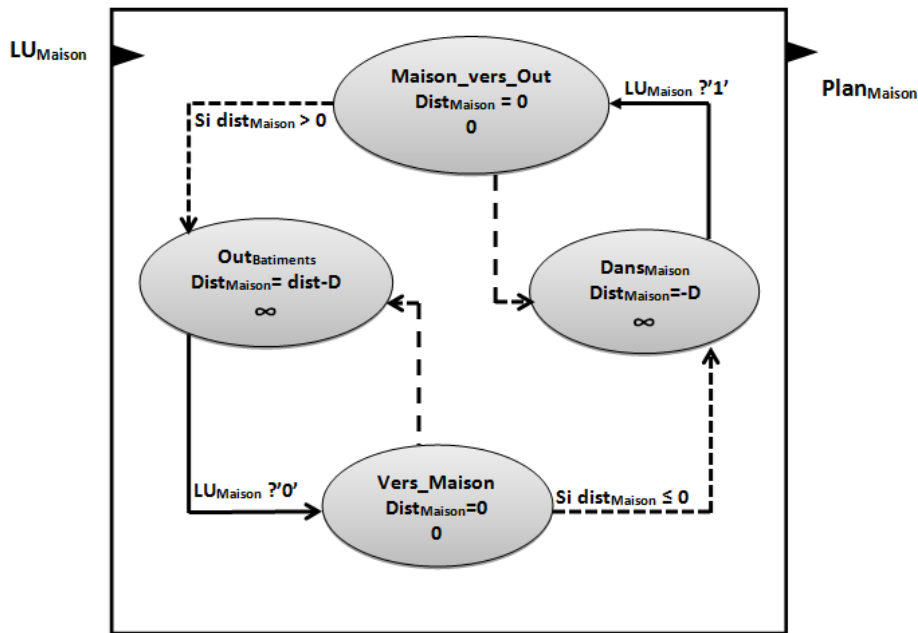


Figure III.9 : Modèle atomique d'un optimiseur

La figure III.9 montre le modèle atomique DEVS de l'optimiseur, avec quatre états : DansMaison, OutMaison, Maison\_vers\_Out, Out\_vers\_Maison. Il y a une entrée pour l'optimiseur : « $LU_{Maison}$ ». La variable  $LU_{Maison}$  (la Localisation de l'Utilisateur par rapport à la maison) prend deux valeurs, «0» ou «1» ; la valeur «0» signifie aller à la maison et la valeur «1» signifie aller à l'extérieur.

La description formelle du modèle «optimiseur» dans DEVS est :

$$\text{Optimiseur} = (X, S, Y, \delta_{\text{ext}}, \delta_{\text{int}}, \lambda, ta) \quad \text{Où:}$$

- $X = \{LU_{\text{Maison}}\}$  présente les variables d'événement d'entrée, où :
- $LU_{\text{Maison}} = \{0,1\}$  Ce port indique l'emplacement de l'utilisateur par rapport à la maison:  
L'événement extérieur  $LU_{\text{Maison}}?'0'$  indique que l'utilisateur se dirige vers la maison;  
L'événement extérieur  $LU_{\text{Maison}}?'1'$  indique que l'utilisateur quitte la maison.
- $Y = \{DANS_{\text{MAISON}}, OUT_{\text{BATIMENTS}}\};$
- L'ensemble des variables d'état:  
 $S = \{\text{phase}, Dist_{\text{Maison}}, ta\}$ , où:
  - $\text{Phase} = \{\text{Dans}_{\text{Maison}}, \text{Out}_{\text{Bâtiments}}, \text{Maison\_vers\_Out}, \text{Vers\_Maison}\};$
  - 'Dist<sub>Maison</sub>', est une variable qui représente la distance actuelle de l'utilisateur par rapport au bâtiment objet (maison), allant de -D à dist, où ( $Dist_{\text{Maison}} \leq 0$ ) représente 'Dans<sub>Maison</sub>', et ( $Dist_{\text{Maison}} > 0$ ) et représente 'Out<sub>Bâtiments</sub>',  $Dist_{\text{Maison}} \in [-D, \text{dist}]$ ;
- $Y = l(Dist)$ , où  $l$  est la fonction de sortie de ce modèle atomique;
- $\delta_{\text{ext}}(DANS_{\text{Maison}}, LU_{\text{Maison}}?'1') = (Maison\_vers\_Out, Dist_{\text{Maison}}=0, 0)$ ,  $\delta_{\text{ext}}(OUT_{\text{Bâtiments}}, LU_{\text{Maison}}?'0') = (Vers\_Maison, Dist_{\text{Maison}}=0, 0)$ ;
- $\delta_{\text{int}}(Vers\_Maison) = DANS_{\text{Maison}}$  si  $Dist_{\text{Maison}} \leq 0$ ,  $\delta_{\text{int}}(Maison\_vers\_Out) = OUT_{\text{Bâtiments}}$  si  $Dist_{\text{Maison}} > 0$ ;  $\delta_{\text{int}}(Maison\_vers\_Out) = DANS_{\text{Maison}}$  si  $Dist_{\text{Maison}} \leq 0$ ;
- $l(Vers\_Maison) = DANS_{\text{MAISON}}$  si  $Dist_{\text{Maison}} \leq 0$ ,  $l(Vers\_Maison) = OUT_{\text{BATIMENTS}}$  si  $Dist_{\text{Maison}} > 0$ ,  $l(Maison\_vers\_Out) = OUT_{\text{BATIMENTS}}$  si  $Dist_{\text{Maison}} > 0$ ;
- $ta(Maison\_vers\_Out) = 0$ ,  $ta(Vers\_Maison) = 0$ ,  $ta(DANS_{\text{Maison}}) = \infty$  et  $ta(OUT_{\text{Bâtiments}}) = \infty$ .

#### III.4.2.4 Modélisation du solveur

On passe à la couche réactive, le solveur, qui est un acteur important de l'optimisation de la géolocalisation pour la gestion énergétique des bâtiments. Le rôle de cet élément est d'ajuster efficacement l'utilisation de l'énergie dans le bâtiment en fonction du schéma d'allocation reçu par l'optimiseur. La figure III.10 montre le modèle atomique DEVS du solveur, avec cinq états : Normal, Vers\_économique, Economique, Vers\_luxe et Luxe.

Le solveur a deux entrées : "plan" et "pas". La variable de plan (déclencheur de luxe/déclencheur d'économie) a deux valeurs, "0" ou "1"; une valeur de "0" signifie déclencher le luxe et une valeur de "1" signifie déclencher l'économie. Ce modèle peut être modifié en statut «luxe», «économique» ou « Normal ».

La description formelle DEVS du modèle «Solveur» est :

$$\text{Solveur} = (X, S, Y, \delta_{\text{ext}}, \delta_{\text{int}}, \lambda, ta) \quad \text{Où :}$$

- $X = \{Plan, pas\}$  présente les variables d'événement d'entrée, où :
  - $Plan = \{0,1\}$  ce port donne l'ordre de déclencher la politique d'utilisation de l'énergie :
  - L'événement extérieur  $Plan?'1'$  indique s'ajuster vers le mode luxe (confort) ;

L'événement extérieur  $Plan?'0'$  indique s'ajuster vers le mode économique. 'pas' est une valeur absolue d'ajustement,  $pas \in [-1, 1]$ .

- $Y = \{ECONOMIQUE, NORMAL, LUXE\}$ ;

- L'ensemble de variables d'état:

$S = \{(phase, m, ta)\}$ , où:

- $Phase = \{Luxe, Normal, Economique, Vers_Luxe, Vers_Economique\}$  ;
- 'm' est une variable qui représente le mode actuel calculé par le solveur,  $m \in \{-1, 0, 1\}$ , où la valeur « -1 » représente le mode économique , la valeur « 0 » représente le mode normal et la valeur « 1 » représente le mode luxe;

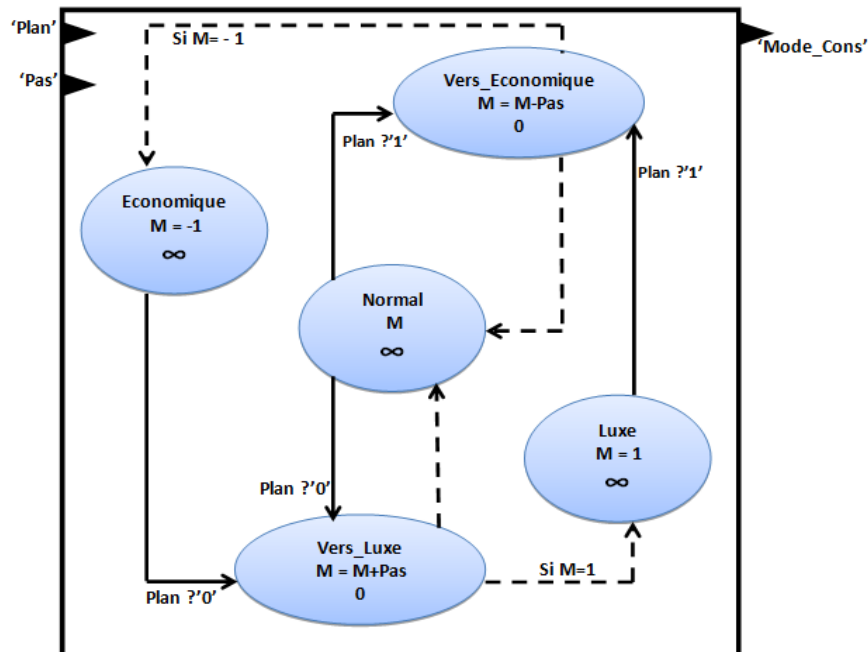


Figure III.10 : Modèle atomique DEVS d'un solveur

- $Y = l(m)$ , où  $l$  est la fonction de sortie de ce modèle atomique;
- $\delta_{\text{ext}}(\text{Economique}, Plan?'0') = (\text{Vers\_Luxe}, m = m + pas, 0)$ ,  $\delta_{\text{ext}}(\text{ordinaire}, Plan?'1') = (\text{Vers\_Economique}, m = m - pas, 0)$ ,  $\delta_{\text{ext}}(\text{ordinaire}, Plan?'0') = (\text{Vers\_Luxe}, m = m + pas, 0)$  et  $d_{\text{ext}}(\text{Luxe}, Plan?'1') = (\text{Vers\_Economique}, m = m - pas, 0)$ ;

- $\delta_{int}(Vers\_Luxe) = \text{ordinaire}$  et  $\delta_{int}(Vers\_Economic) = \text{Ordinaire}$  si  $-1 < m < 1$ ,  
 $\delta_{int}(Vers\_Economic) = \text{Economic}$  si  $m \leq -1$ ,  $\delta_{int}(Vers\_Luxe) = \text{Luxe}$  si  $m \geq 1$ ;
- $l(Vers\_Economiq\ue) = \text{ECONOMIQUE}$  si  $m = -1$ ,  $l(Vers\_Economiq\ue) = \text{ORDINAIRE}$   
 si  $-1 < m \leq 0$ ,  $l(Vers\_Luxe) = \text{ORDINAIRE}$  si  $0 \leq m < 1$ , et  $l(Vers\_Luxe) = \text{LUXE}$   
 si  $m = 1$ ;
- $ta(Vers\_Luxe) = 0$ ,  $ta(Vers\_Economiq\ue) = 0$ ,  $ta(\text{Ordinaire}) = \infty$ ,  $ta(\text{Economiq\ue}) = \infty$   
 et  $ta(\text{Luxe}) = \infty$ .

### III.4.2.5 Modélisation du régulateur

L'appareil fonctionne en temps réel et la couche native est la couche qui utilise le sous-système régulateur pour adapter la portée du mécanisme réactif à l'appareil. A travers le régulateur, le système impose des commandes sur la consommation énergétique de l'équipement, par les actions suivantes :

- Application du contrôle réactif de la consommation des équipements.
- Affinement de l'horizon.

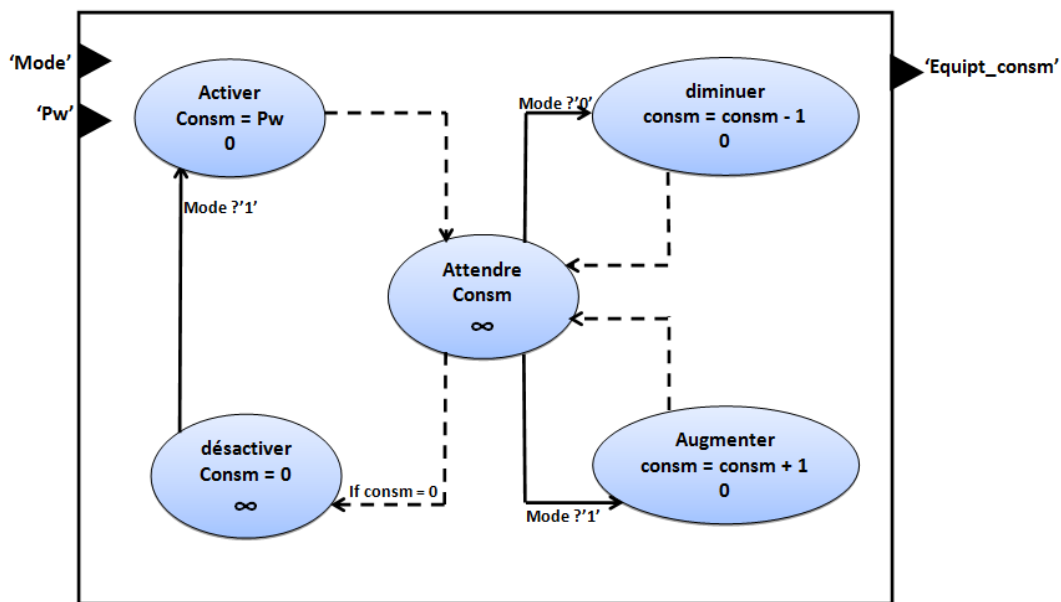


Figure III.11 : Modèle atomique DEVS d'un régulateur

Le régulateur contrôle la consommation d'énergie des équipements temporaires avec des procédures de démarrage ou d'arrêt et contrôle également les équipements permanents en augmentant ou en diminuant leur consommation d'énergie. Il existe deux valeurs d'entrée, 0 et 1 (figure III.11). La valeur 1 augmente la consommation de l'équipement

concerné et la valeur 0 la réduit. Le fonctionnement du régulateur est exprimé en pourcentage de la puissance opérationnelle (par exemple, 0% = arrêt, 100% = puissance de pointe). La consommation de l'équipement concerné peut varier de zéro à la puissance maximale (Pw).

La description formelle DEVS du modèle «régulateur» est :

$$\text{Régulateur} = (X, S, Y, \delta_{ext}, \delta_{int}, \lambda, ta) \quad \text{Où:}$$

- $X = \{Mode, Pw\}$  présente les variables d'événement d'entrée, où :
  - $Mode = \{0,1\}$ , ce port indique la politique d'utilisation de l'énergie à appliquer :

L'événement extérieur  $mode? '1'$  augmente la consommation de l'équipement permanent et si le type de l'équipement = temp, cet évènement allume l'équipement temporaire; L'événement extérieur  $mode? '0'$  diminue la consommation de l'équipement.

- 'Pw' est une variable qui indique la consommation maximale de l'équipement.
- $Y = \{ACTIVER, DESACTIVER, AUGMENTER, DIMINUER\}$ ;
- L'ensemble de variables d'état:  $S = \{(phase, consm, ta)\}$ , où:
  - $Phase = \{Activer, Désactiver, Augmenter, Diminuer, Attendre\}$ ;
  - 'consm' est une variable qui indique la consommation actuel de l'équipement allant de 0 à Pw,  $consm \in [0, Pw]$ ;
  - $Y = \lambda (consm)$ , où  $\lambda$  est la fonction de sortie de ce modèle atomique;
  - $d_{ext} (Désactiver, mode? '0') = (Activer, consm = Pw, 0)$ ,  $d_{ext} (Attendre, mode? '1') = (Augmenter, consm = consm + 1, 0)$  et  $d_{ext} (Attendre, mode? '0') = (Diminuer, consm = consm - 1, 0)$ ;
  - $\delta_{int} (Augmenter) = Attendre$ ,  $\delta_{int} (Diminuer) = Attendre$ ,  $\delta_{int}(Activer) = Attendre$  et  $\delta_{int} (Attendre) = Désactiver$  si  $consm = 0$ ;
  - $\lambda(Activer) = consm*50/Pw$ ,  $\lambda(Augmenter) = consm*75/Pw$ ,  $\lambda(Diminuer) = consm*25/Pw$ ;
  - $ta(Désactiver) = \infty$ ,  $ta(Activer) = 0$ ,  $ta(Attendre) = \infty$ ,  $ta(Diminuer) = 0$  et  $ta(Augmenter) = 0$ .

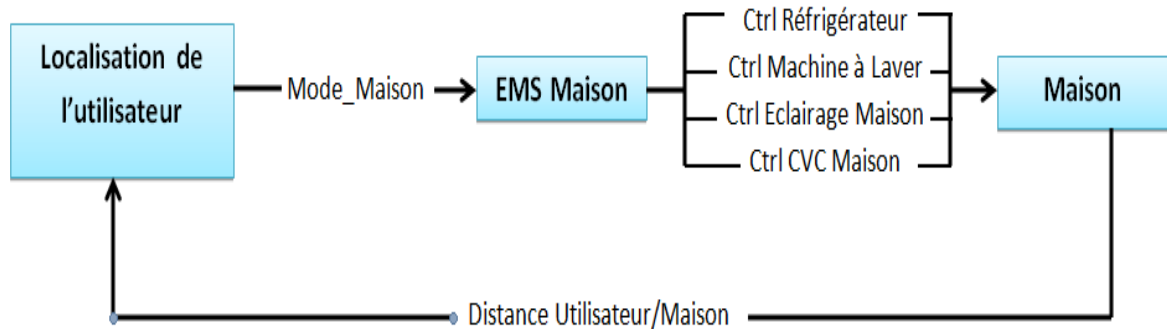
### III.4.3 Le modèle de simulation DEVS

La figure III.12 présente le modèle DEVS couplé d'un environnement formé par un utilisateur qui se déplace chaque jour de sa maison vers l'extérieur et revient après avoir terminé son travail. Cet environnement correspond à la mise en œuvre du modèle illustré dans



la figure III.6. Le modèle couplé EMS Maison et le modèle Localisation de l'utilisateur ont été modélisés séparément pour former le problème de simulation correspondant à cet environnement.

La simulation commence lorsque l'utilisateur quitte son domicile. Ceci est confirmé lorsque l'utilisateur franchit un seuil bien défini. Le service de localisation de l'utilisateur envoie immédiatement un message de décision de mode à l'EMS dans le bâtiment de maison, qui apporte ensuite les modifications nécessaires à la politique d'économie d'énergie en allumant / éteignant les appareils électroménagers de ce bâtiment.



**Figure III.12 : Modèle de simulation DEVS pour la gestion de l'énergie dans la Maison**

### **III.5 Conclusion**

Dans ce chapitre, nous avons conçu notre modèle de simulation DEVS pour la gestion de l'énergie dans un environnement qui comporte une maison et un utilisateur accompagné avec son Smartphone. Son objectif est de permettre d'illustrer l'optimisation de la consommation d'énergie via des simulations de modèles DEVS de manière générique et performante. La description de cette conception s'est déroulée en trois étapes.

Dans un premier temps, nous isolons les différentes entités qui composent notre environnement applicatif et élucidons leurs apports en décrivant des algorithmes pouvant faciliter l'optimisation de la consommation d'énergie dans cet environnement. Dans un deuxième temps, nous concevons un modèle de gestion énergétique du bâtiment intelligent multi-niveaux qui respecte l'architecture décrite dans le travail présenté dans [Maato, 2014]. La structure porte plusieurs niveaux de modélisation, correspondant à plusieurs niveaux de commandes, divisant le problème d'optimisation en sous-problèmes afin d'être plus réactif à l'incertitude. Dans la troisième et dernière étape, nous formalisons tous les modèles proposés dans DEVS. Pour cela, nous choisissons de baser notre modélisation sur des représentations graphiques et des descriptions formelles.

Une fois ces trois étapes terminées, nous passons à l'étape d'implémentation et de validation des différents modèles DEVS dans différents scénarios. Ceci est décrit dans le chapitre suivant.

# **CHAPITRE IV**

## **Simulation et expérimentation**

## **Chapitre IV : Simulation et expérimentation**

### **IV.1 Introduction**

Les systèmes présents dans notre environnement d'application peuvent être considérés comme des structures complexes. Le comportement et les interactions dans ces systèmes sont très compliqués. Les principes de modularité et de décomposition hiérarchique énoncés dans notre présentation du formalisme DEVS nous permettent d'adopter un niveau de détail suffisant pour exprimer cette complexité. Plusieurs simulateurs sont associés au formalisme DEVS. Ces simulateurs sont des environnements basés sur des langages de programmation plus évolués comme C++, Java, ou autre, ce qui donne beaucoup d'avantages au modélisateur et surtout la possibilité de modifier ou de reconfigurer dans son modèle. Pour nous, notre choix d'outil pour développer et simuler nos modèles s'est porté sur JDEVS. Ce chapitre présente un aperçu des modèles développés à l'aide de l'environnement de modélisation et de simulation JDEVS. L'objectif poursuivi lors de ce développement était de satisfaire le besoin d'un outil de modélisation simple d'utilisation tout au long de notre travail. JDEVS est largement utilisé dans divers projets de recherche et s'est avéré efficace pour valider les conceptions et démontrer leur dynamique.

À l'exception de cette introduction, ce chapitre est divisé en quatre sections, la première section décrit l'outil de simulation JDEVS, la deuxième section détaille l'implémentation de tous les modèles, la troisième section traite des résultats expérimentaux et la dernière section compare notre méthodologie avec un travail récent plus similaire au nôtre.

## IV.2 JDEVS : notre environnement de modélisation et simulation

Pour évaluer les performances de l'approche proposée, nous avons implémenté et simulé tout ce qui existe dans l'environnement d'application avec le simulateur JDEVS [Filip, 2004]. JDEVS est un outil qui implémente les modèles DEVS en code Java pour spécifier leur comportement. JDEVS est composé de cinq modules indépendants représentés sur la figure IV.1 (les carrés correspondent aux modules, les losanges correspondent aux acteurs et les cercles aux formats d'échange de données) : un moteur de modélisation/simulation, une interface graphique de modélisation, un module de stockage en librairies, un module pour les connexions SIG et les cadres expérimentaux pour la visualisation et la simulation.

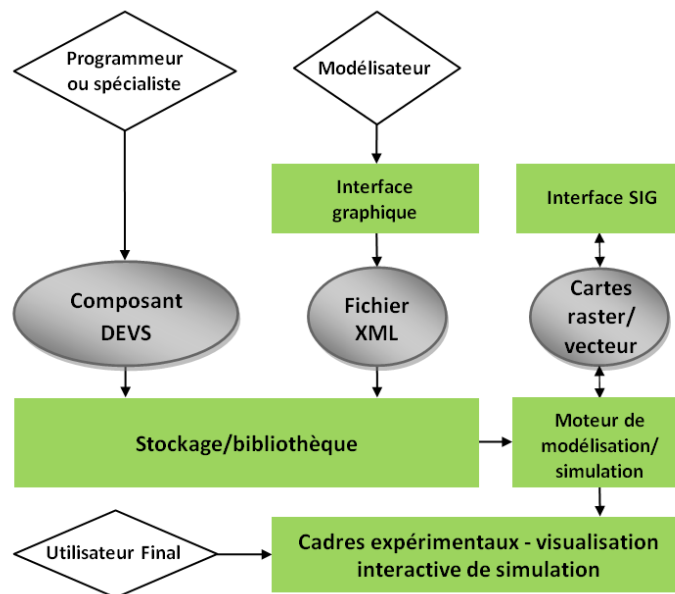


Figure IV.1 : Vue des modules logiciels JDEVS

Le « Moteur de modélisation et de simulation » est une implémentation de la méthode DEVS classique avec des ports. Dans le processus d'implémentation, le modèle DEVS atomique sera immédiatement converti en instructions JAVA. Outre le fait que le modèle atomique peut être conçu directement dans l'interface à l'aide de code Java, le formalisme DEVS permet également l'utilisation de modèles qui sont programmés dans plusieurs langages orientés objet via des appels à distance, tout comme l'interface de programmation Java RMI<sup>3</sup>. De plus, lors de l'ajout d'un composant atomique vide, après avoir inséré son

<sup>3</sup> Java RMI (Remote Method Invocation): une interface de programmation (API) pour le langage Java qui permet d'appeler des méthodes distantes.

interface d'entrée et de sortie, le squelette de code du modèle sera automatiquement généré, et le modèle atomique sera enregistré dans la bibliothèque et compilé.

Le code généré par le modèle atomique  $MA = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta \rangle$  est le suivant :

```
public class DevsAtom extends BasicModel {
    Port i1 = new Port(this, "i1", "IN") ;
    Port o1 = new Port(this, "o1", "OUT") ;

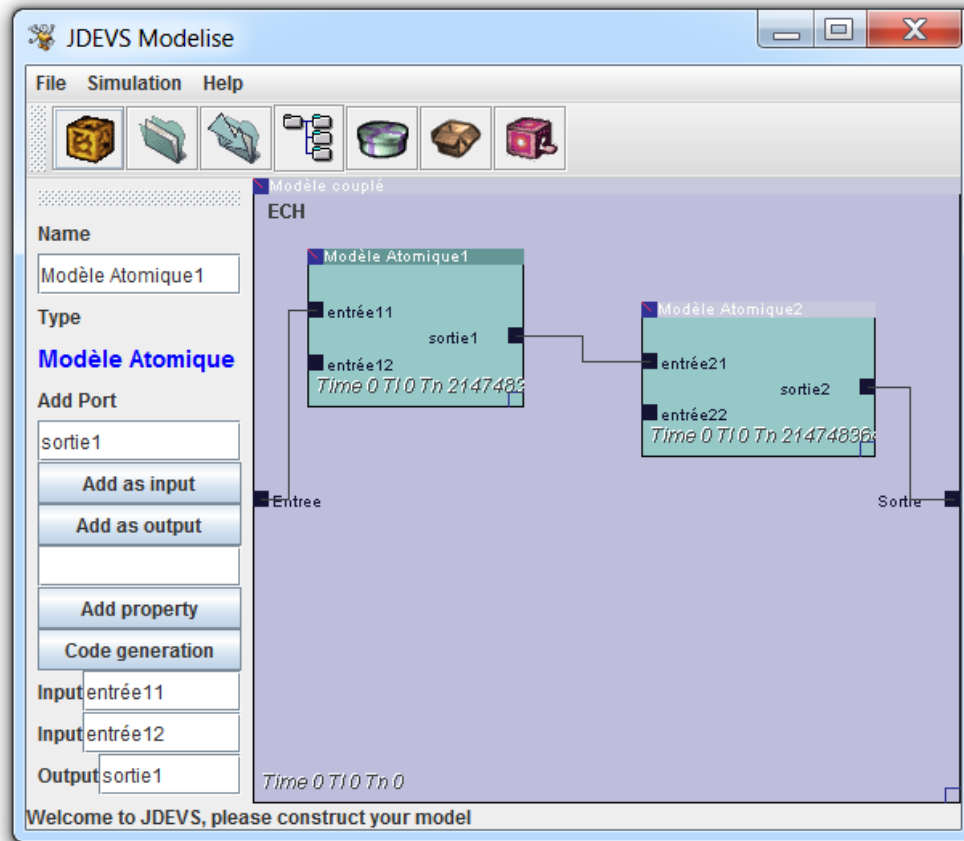
public DevsAtom () {
super("DevsAtom") ;
states.setProperty("A", "") ; }
    EventVector outFunction(Message m) {
        return new EventVector() ;}
    void intTransition() {}
    EventVector extTransition(Message m) {
        return new EventVector() ;}
    int advanceTime() {return A ;}
}
```

Les fonctions de sortie et de transition externe renvoient de nouveaux vecteurs d'événements qui sont ensuite stockés dans la liste générale des événements. Notre seule tâche de programmation est de déterminer la dynamique des modèles atomiques par ses quatre méthodes. Lorsque le modèle atomique est créé, il sera simultanément stocké dans la bibliothèque pour être utilisé dans la construction d'un modèle couplé prédéterminé.

La construction graphique de modèles se fait dans «l'interface graphique de modélisation». Cette interface se compose des éléments suivants (voir dans la figure IV.2) :

- Un espace de travail (centre de la figure IV.2) pour construire les modèles ;
- Un panneau de propriétés (à gauche de la figure IV.2) lié à chaque modèle permettant de modifier la valeur de ses paramètres, d'ajouter et de supprimer des ports et de changer son nom, de lancer l'éditeur de code Java et de compiler ;
- Une barre de menu (haut de la figure IV.2) permettant d'ajouter des modèles vides (atomique, couplés), d'ouvrir une bibliothèque et de supprimer des modèles ;

Les ports sont représentés par une poignée noire dans un modèle composant ; les couplages s'effectuent en cliquant dans les ports source puis destination. Les composants peuvent être déplacés et redimensionnés avec la souris. Nous pouvons également ajouter des modèles existants en les faisant glisser depuis la bibliothèque.

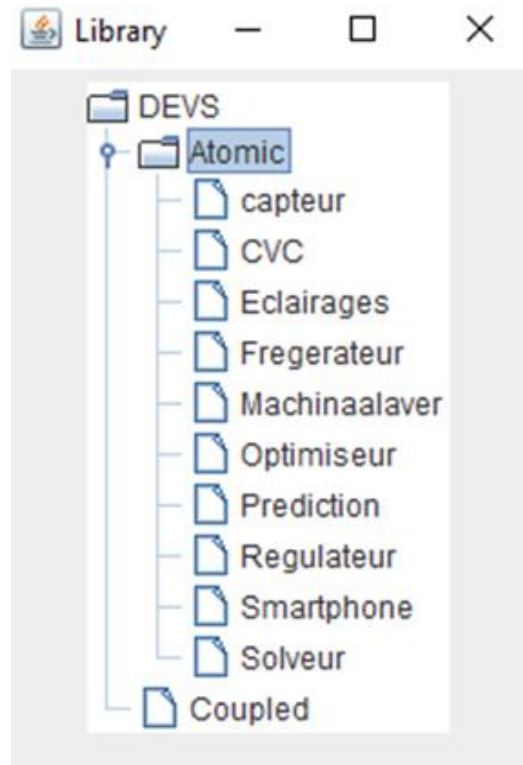


**Figure IV.2 : Interface de modélisation d'un modèle couplé avec son panneau de propriétés**

Le « module de stockage » de JDEVS est un composant graphique présenté en figure IV.3; Il suffit de cliquer sur son icône dans la barre de menu dans l'interface pour l'ouvrir. Ce composant est une vue arborescente des modèles par domaine et sous-domaine. Son objectif est de fournir un moyen simple et efficace pour stocker et récupérer les modèles.

Le module de stockage permet d'enregistrer les modèles en format XML, et de les stocker dans une bibliothèque HmLib<sup>4</sup> [Berna. 2002].

<sup>4</sup>La bibliothèque HmLib peut être considérée comme une base de données orientée objet, ce qui facilite le stockage des modèles en XML.



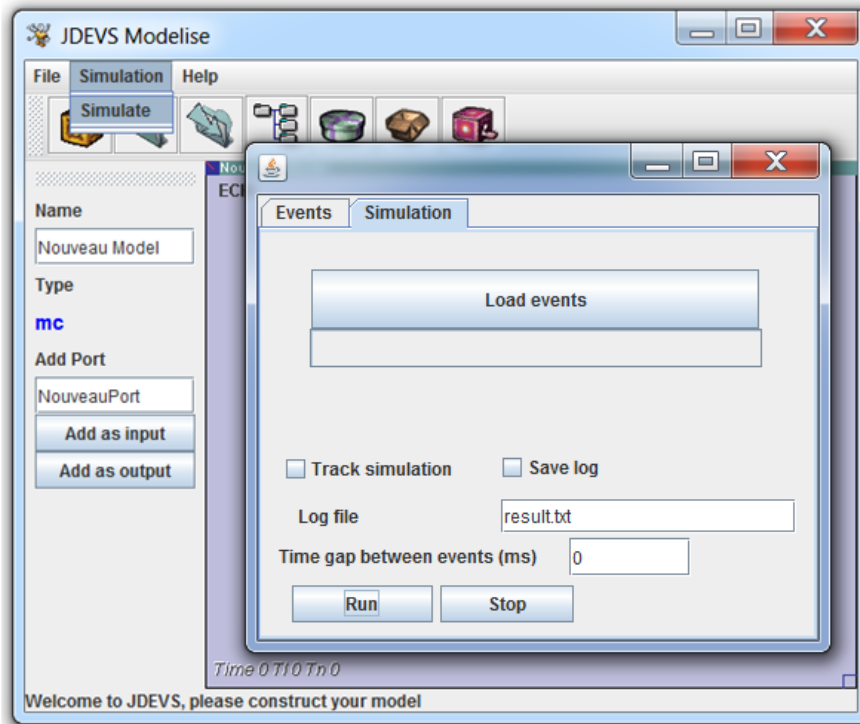
**Figure IV.3 : Composant de l'interface graphique présentant les modèles dans la bibliothèque**

Ainsi, Il est possible de stocker les liens d'héritage et d'abstraction entre les modèles. Les modèles stockés dans la bibliothèque sont initialement en mode "hors-contexte". Pour récupérer un modèle, il faut l'instancier puis le mettre "en-contexte" en le replaçant dans l'état où avait été stocké. Dans les cadres expérimentaux, la simulation peut être lancée qu'à partir de modules "en-contexte" [Filip, 2004]

Les « cadres expérimentaux » sont des interfaces qui permettent de simuler les modèles créés. Les expériences sur les modèles peuvent être effectuées directement à partir de l'interface graphique de modélisation. Pour exécuter la simulation, nous téléchargeons la liste des événements d'entrée et définissons les paramètres de l'expérience via le panneau de simulation (Figure IV.4).

Pour observer le comportement du modèle, il suffit d'activer l'option "track simulation" et de saisir un temps d'attente entre l'envoi de deux événements d'entrée. Lors de la simulation, la liste d'événements (à traiter) est affichée à l'intérieur du modèle couplé et évolue au cours de la simulation. Les résultats de la simulation peuvent être affichés ou enregistrés dans un fichier texte.





**Figure IV.4 : Comment effectuer des simulation dans l'environnement JDEVS**

Il existe plusieurs méthodologies d'« interconnexion au SIG » [Brand, 2000] :

- l'intégration totale, dont l'environnement de simulation est développé avec un module SIG où les données sont modifiées directement lors de la simulation,
- le couplage "serré" où chaque modification des données entraîne une mise à jour de la base de données lors de la simulation,
- le couplage "lâche" où les données d'initialisation sont importées du modèle et les résultats sont exportés vers le SIG à la fin de la simulation.

Puisque lors du développement de JDEVS, l'objectif était que JDEVS soit indépendant de tout logiciel tiers, et sous prétexte que le couplage "serré" ralentit la simulation, le couplage "lâche" a été opté [Filip, 2004]. Contrairement à l'importation, l'exportation des résultats de simulation est plus compliquée car le SIG ne peut pas stocker les attributs temporels. La carte n'est donc pas complètement remise à jour à intervalles réguliers. Pour permettre l'exportation des résultats, il est nécessaire de recréer des cartes raster à partir d'événements en sortie. Il est donc impératif de choisir l'intervalle requis entre deux cartes de sortie, de sorte que le module de connexion au GIS les reconfigure en appliquant toutes les modifications depuis l'initialisation.

### IV.3 Simulation du modèle proposé à l'aide de l'outil JDEVS

Dans DEVS, un cadre expérimental est utilisé pour effectuer des tests de validation. Si le comportement du modèle et de son homologue système se situe dans les limites d'une tolérance acceptable, le modèle est valide [Zeigl, 2000] Pour ce faire, nous avons modélisé tous les sous-systèmes de l'environnement décrit ci-dessus selon le formulaire DEVS, et ci-dessous nous plaçons ces modèles sous l'outil de simulation JDEVS. De cette façon, nous pouvons exécuter des expériences sur nos modèles afin d'étudier leur efficacité et leurs performances.

#### IV.3.1 Représentation du modèle global à l'aide de l'outil JDEVS

Nous commençons maintenant l'étape d'installation et d'exécution de tous nos modèles couplés et atomiques. Pour ce faire, nous utilisons, l'environnement de modélisation et simulation JDEVS qui a été expliqué dans la section précédente.

Pour rappel, notre scénario d'application comprend un utilisateur d'un bâtiment (domicile), accompagné de son smartphone. Dans notre système, ce qui représente la maison est le système de gestion de l'énergie EMS\_Maison, Par conséquent, le modèle global comprenait le modèle couplé EMS\_Maison, et le modèle atomique du Smartphone (Figure IV.5).

Il s'agit d'un modèle générique qui peut représenter divers types d'appareils

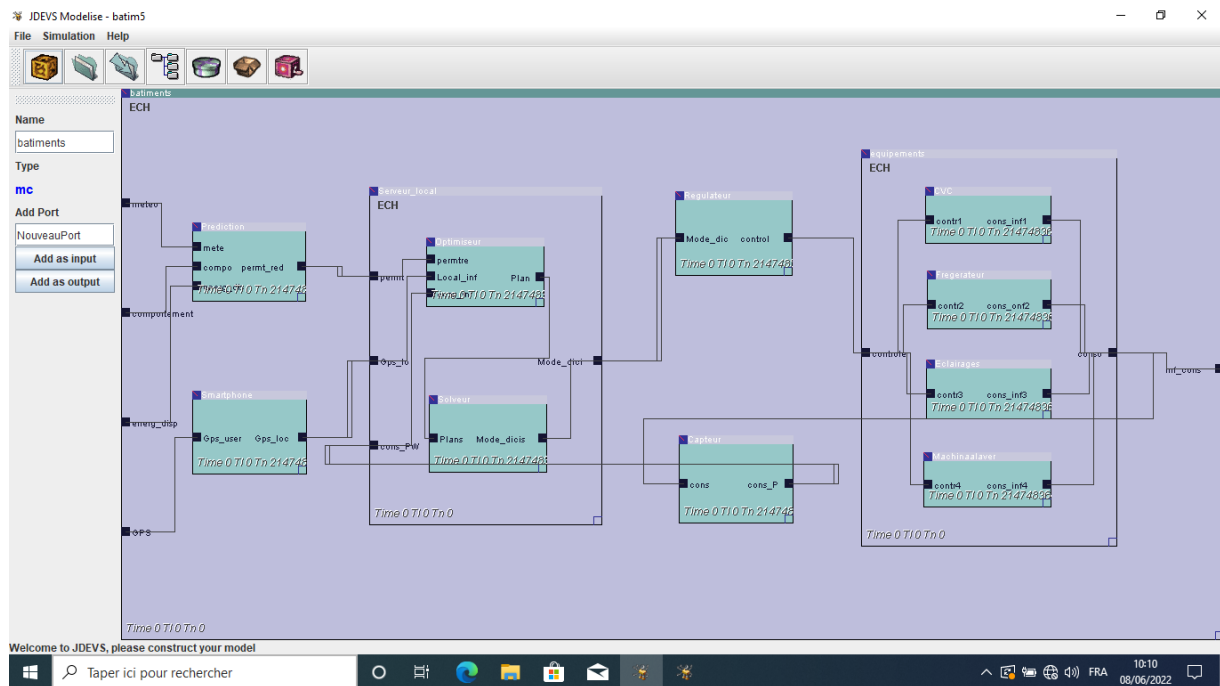


Figure IV.5 : Représentation du modèle global à l'aide de l'outil JDEVS

électroménagers et d'autres appareils au sein d'un système de gestion de l'énergie d'un bâtiment, ainsi qu'un sous-système atomique qui représente le smartphone de l'utilisateur de ce bâtiment. La façon dont ces sous-systèmes composants sont modélisés en fonction de leur comportement est similaire, mais la fonctionnalité est différente. Dès le début de la modélisation jusqu'à la phase d'implémentation, nous avons soigneusement étudié les fonctions attribuées à chaque composant du système, et après avoir programmé chaque modèle atomique en Java séparément, nous avons pu atteindre la dynamique attendue des modèles couplés jusqu'à ce que nous ayons terminé le modèle global.

Ce modèle se compose de plusieurs modèles atomiques. Si nous entrons le temps de mise en œuvre pour chaque modèle, notre modèle est divisé en trois couches, chacun est soumis à une échelle temporelle différente [Maato, 2014]. La couche d'anticipation contient les modèles atomiques *Smartphone de l'utilisateur*, *Prédictions* et *Optimiseur*, la couche réactive contient *Solveur* et *Regulateur* et la couche locale contient *Capteurs et Appareils ménagers* représentés par : *Fridge*, *Wash\_Mach*, *Light* et *CVC*.

Au départ, lorsque les utilisateurs du système connectent un appareil au système, l'appareil envoie des messages d'identification qui incluent le nom et l'ID à un modèle d'identification. Si l'authentification est correcte, le système commence à fonctionner. En outre, lors de la mise en œuvre, nous avons dû ajuster les temps d'échantillonnage pour chaque couche. Un planning de fonctionnement est joint pour chaque couche du mécanisme de l'EMS. La couche anticipative travaille sur un horizon temporel de l'ordre d'une heure, la couche réactive de l'ordre d'une minute et la couche locale travaille en temps réel.

L'expérimentation de nos modèles peut être réalisée directement à partir de l'interface graphique de modélisation. Par conséquent, le comportement du modèle en cours de création, peut être vérifié pour ajuster ses paramètres ou corriger les erreurs. En outre, l'application utilisée pour exécuter les expériences existe dans la même interface de modélisation.

En général, l'objectif principal de la validation est de vérifier si un système fonctionne comme prévu, dans différentes conditions et scénarios de simulation. En d'autres termes, la validation vise à rendre le modèle utile dans le sens où il répond à un bon problème.

Notre problème a été détaillé et formulé dans la section III.2.2 du chapitre précédent, et dans la section suivante, nous examinons les performances de cette approche dans différents scénarios de simulation.

## IV.4 Résultats expérimentaux et discussion

Afin de clarifier le potentiel de cette approche, nous avons dû réfléchir attentivement à une stratégie qui prouve son efficacité. Nous avons exploré les facteurs qui ont eu le plus d'influence sur le façonnement des scènes, les principaux acteurs, notamment les modes de consommation d'énergie, les lieux de présence potentielle de l'utilisateur et les événements critiques qui conduisent à des changements de situations. Par conséquent, nous avons réalisé les étapes décrites dans les sous-sections suivantes :

### IV.4.1 La quantité d'énergie estimée dans différent modes de consommation

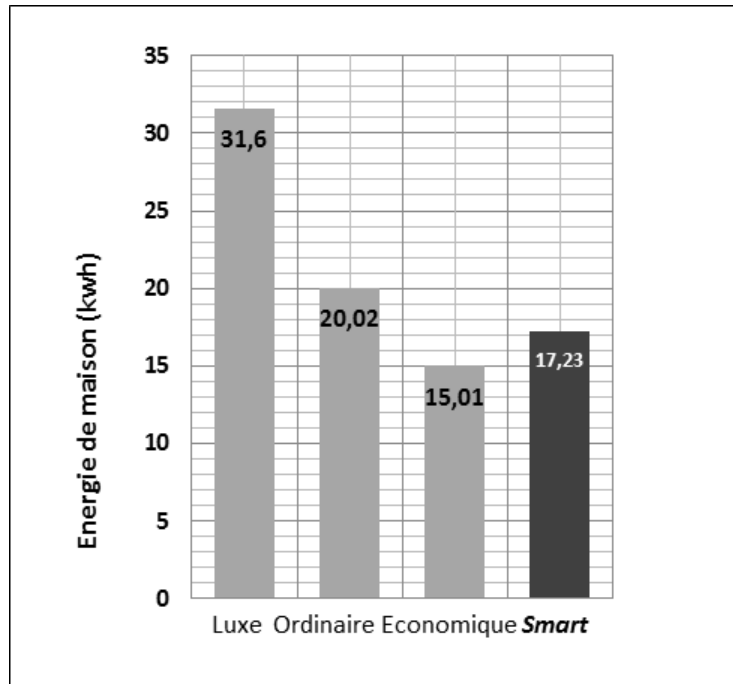
Avant de commencer les simulations, nous avons dû expliquer les trois modes de consommation d'énergie : le mode Économique, Ordinaire et Luxe, en montrant les quantités d'énergie consommées par tous les appareils électriques dans chaque mode pour la maison.

Nous montrons dans le tableau IV.1, les résultats de l'estimation de la consommation d'énergie quotidienne pour la maison [Badre, 2012], illustrant ces trois modes.

	Éclairage	Réfrigérateur	Machine à laver	Autres appareils	CVC
<b>Mode Luxe (***)</b>	8000 wh	2100 wh	2000 wh	3000 wh	16500 wh
<b>Mode Ordinaire (**)</b>	4000 wh	2100 wh	1000 wh	2400 wh	10518 wh
<b>Mode Economique (*)</b>	2400 wh	2100 wh	600 wh	1500 wh	8410 wh
<b>Total</b>	(***) : 31.60 kWh ; (**): 20.02 kWh; (*): 15.01 kWh.				

**Tableau IV.1 : Estimation de la consommation énergétique quotidienne de la maison**

Ensuite, nous changeons l'emplacement de l'utilisateur en supposant qu'il passe 08 heures à travailler, 08 heures à l'intérieur de la maison et 08 heures à l'extérieur. Nous appliquons un contrôle dynamique et des changements de politique à la maison au moyen de simulations et nous enregistrons les résultats de la consommation d'énergie dans les 24 heures. Nous appelons ce mode de consommation d'énergie le «mode smart» et nous le comparons aux trois modes virtuels : économique, ordinaire et luxe (Voit Figure IV.6).



**Figure IV.6 : Consommation d'énergie de la maison avec l'estimation des trios modes**

La figure IV.6 montre cette comparaison avec précision. D'après ce que montre cette figure, nous avons remarqué qu'après avoir appliqué l'ajustement rationnel de la consommation d'énergie à la maison, nous consommons juste la quantité d'énergie équivalente à ce qu'elle est en mode économique en préservant tout ce que le mode luxe apporte de confort. De cette façon, nous avons réduit la consommation d'énergie au maximum, et sans préjudice des exigences de bien-être de l'utilisateur.

#### **IV.4.2 Les données de simulation**

Afin de prouver que l'optimisation géo localisée de l'utilisateur est à l'origine de cette réalisation, nous surveillons les événements par lesquels l'utilisateur change de position par rapport à l'immeuble de la maison.

En d'autres termes, nous n'enregistrerons que les événements qui conduisent à une différence de distance entre l'utilisateur du bâtiment et le bâtiment par rapport à la distance prédéterminée pour les limites de ce bâtiment (dans nos procédures, nous avons considéré que la distance spécifique au bâtiment égal à 05 mètres par défaut).

Pour cela, nous effectuons des simulations des événements suivants (voir Tableau IV.3).

Date	Distance Maison/Utilisateur	Éclairage	Réfrigérateur	Machine à laver	CVC
18 mai 2022 12:06:49 (temps 0)	12,75	1452	716	0	634
18 mai 2022 12:09:31 (temps 1)	0,00	1508	1027	6	782

Tableau IV.3 : Données de simulation de maison

### IV.4.3 Un 1<sup>ier</sup> scénario : la consommation d'énergie d'un équipement temporaire

Observons maintenant, par exemple, la consommation énergétique de l'éclairage dans toutes les situations où l'utilisateur est présent, tant à l'intérieur qu'à l'extérieur du bâtiment. C'est l'un des services temporaires qui sont fournis dans la maison. La figure 7.IV montre la consommation quotidienne d'énergie d'éclairage pour ce bâtiment.

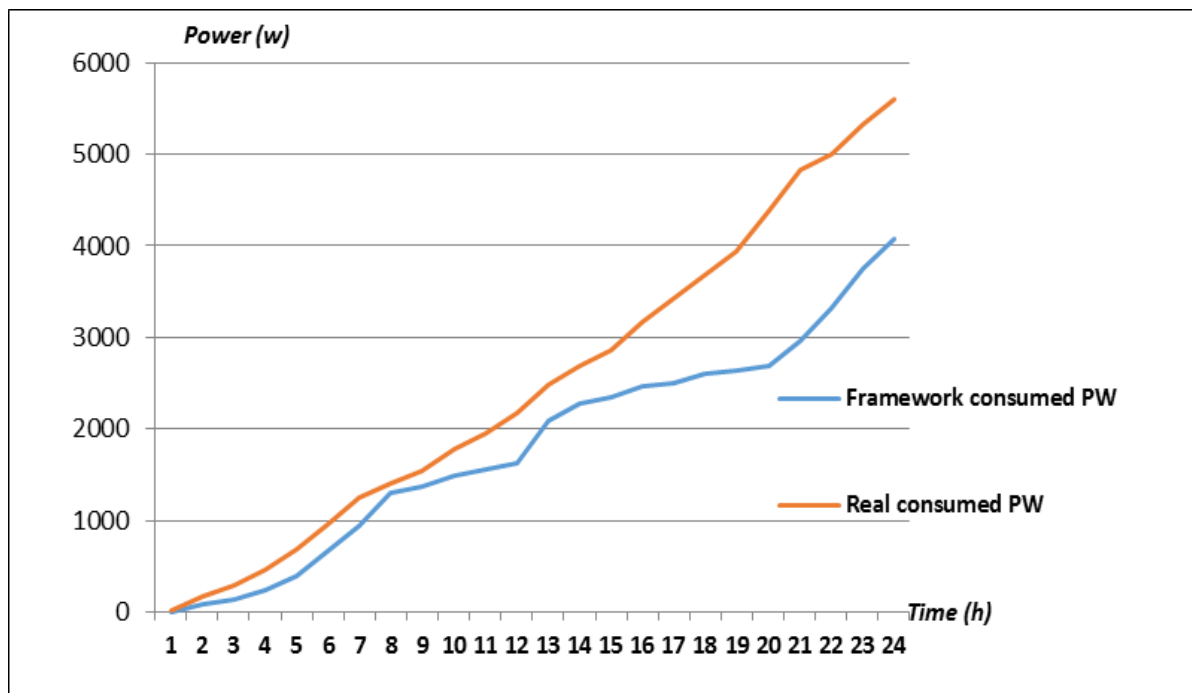


Figure IV.7 : Consommation d'énergie quotidienne de l'éclairage de la maison

A travers cette expérience, nous expliquons comment le système contrôle l'éclairage de bâtiment, par exemple, en surveillant l'utilisateur sur deux distances, l'une d'elles est inférieure

à la distance minimale spécifiée pour le périmètre de la maison qui est égale à 5 mètres, et la seconde est supérieure à cette distance. Dans ces deux positions de l'utilisateur, la quantité d'énergie consommée par l'éclairage dans ce bâtiment peut être relevée dans le tableau IV.1.

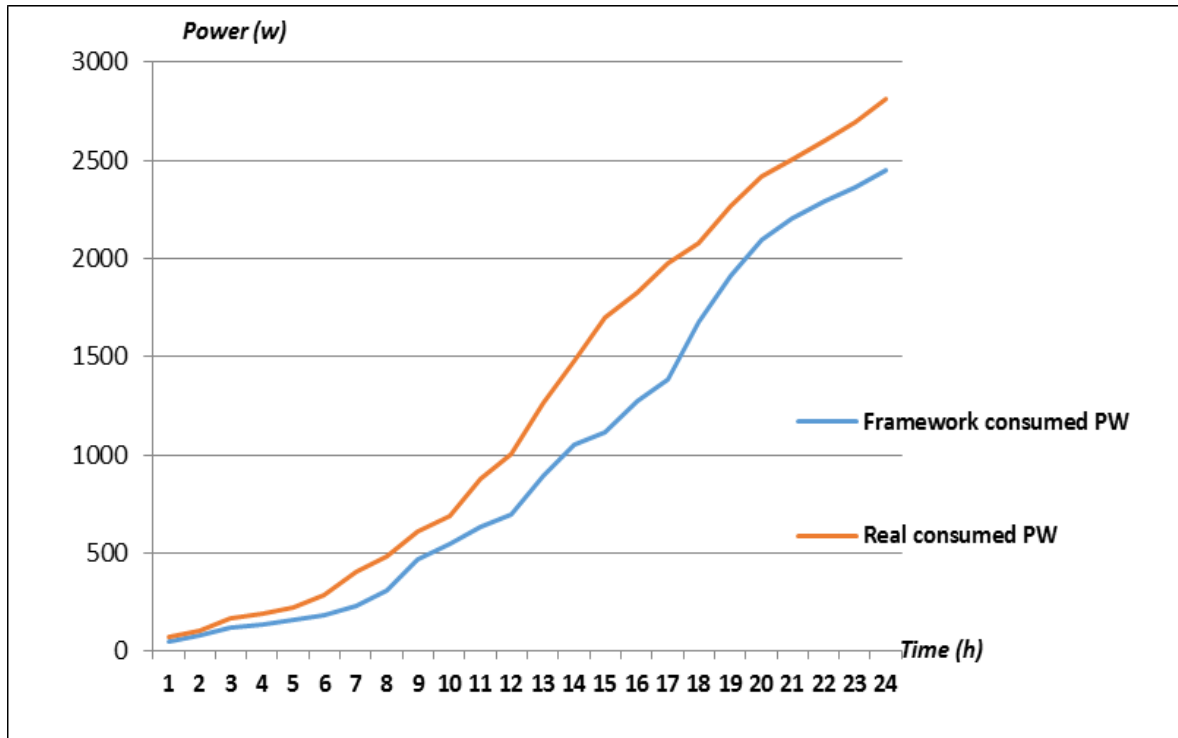
Dans notre travail, nous considérons que le système d'éclairage est constitué d'un groupe de lampes simples et leur contrôle se fait de manière groupée, c'est-à-dire qu'il n'y a que deux cas pour ce système : allumer ou éteindre l'éclairage. Comme nous l'avons expliqué dans le chapitre précédent, spécifiant le modèle atomique des équipements temporaires (voir la section III.4.2.1), une lampe simple ne peut être que dans l'état allumée ou éteinte, et son modèle atomique se compose de quatre états : Off, Essai-on, On et Essai-off. La phase initiale du modèle 'lampe' est «Off».

Au temps 0, la distance entre l'utilisateur et le bâtiment dépassait la valeur minimale ( $\text{Distance} < 5$ ). Juste au moment où la distance minimale est dépassée, le système allume l'éclairage du bâtiment, c'est-à-dire que lorsque le modèle 'lampe' reçoit l'événement «in» sur son port '1', il passe en phase «On». Cette transition dans les cas a été effectuée par la fonction de transition externe. Au temps 1, la distance entre l'utilisateur et le bâtiment est supérieure à la valeur minimale ( $\text{Distance} > 5$ ). Au moment où l'utilisateur dépasse le périmètre de bâtiment, le système éteint l'éclairage de bâtiment, c'est-à-dire que lorsque le modèle de lampe reçoit l'événement «in» sur son port '0', il passe en phase «off». Cette transition dans le cas a également été effectuée par la fonction de transition externe.

#### **IV.4.4 2<sup>ème</sup> scénario : la consommation d'énergie d'un équipement permanent**

Observons maintenant la consommation d'énergie du réfrigérateur aussi dans toutes les situations de présence de l'utilisateur, que ce soit à l'intérieur ou à l'extérieur de la maison. Le réfrigérateur est un service thermique permanent. La figure IV.8 illustre la consommation d'énergie quotidienne du réfrigérateur dans la maison.

A travers cette expérience, nous expliquons comment le système contrôle le fonctionnement de cet équipement en surveillant l'utilisateur sur deux distances, l'une d'elles est inférieure à la distance minimale spécifiée pour le périmètre de la maison qui est égale à 5 mètres, et la seconde est supérieure à cette distance. Dans ces deux positions de l'utilisateur, la quantité d'énergie consommée par un réfrigérateur dans la maison peut être relevée dans le tableau IV.1.



**Figure IV.8 : Consommation d'énergie quotidienne du réfrigérateur**

Lors de l'implémentation, nous avons inclus une fonction qui calcule la différence des distances de déplacement de l'utilisateur calculées pour deux instants consécutifs avec une petite différence de pas de temps (1 seconde). Le but de cette fonction est d'estimer l'orientation de l'utilisateur afin de faire intervenir en amont le système dans la situation de consommation d'énergie en profitant de l'opportunité de réduire ou d'augmenter la consommation d'énergie des équipements permanents.

En d'autres termes, par exemple, si l'utilisateur se dirige vers la maison, le système augmente progressivement la consommation d'énergie des équipements permanents de la maison, et fait l'inverse dans le cas inverse.

Comme nous l'avons expliqué dans le chapitre précédent, spécifiant le modèle atomique des équipements permanents (voir dans la section III.4.2.2), Le réfrigérateur est un équipement permanent et il est régulé de la manière suivante : il est soit à l'arrêt (%P = 0 %), en bas (%P = 25%), en moy (%P = 50%), en haut (%P= 75%), ou soit à la puissance maximale (%P = 100%). Son modèle peut être changé en état On, Off, Essai\_On, Essai\_Off, Attente, Augmenter et Diminuer. La phase initiale du modèle 'Réfrigérateur' est «Off». Au temps 0, la distance entre l'utilisateur et sa maison dépassait la limite supérieure du périmètre de maison (Distance > 5). Lorsqu'il détecte que l'utilisateur s'est dirigé vers la maison, le



système ajuste progressivement le mode de consommation d'énergie du réfrigérateur pour le mettre en mode luxe au moment où l'utilisateur a effectivement atteint le périmètre de la maison, c'est-à-dire que lorsque le modèle 'Réfrigérateur' reçoit l'événement «CCE» sur son port '1', il passe en phase «Augmenter». Cette transition dans les cas a été effectuée par la fonction de transition externe.

Au temps 1, la distance entre l'utilisateur et sa maison est égale à la valeur spécifiée pour le périmètre de la maison (Distance = 5). Au moment où l'utilisateur atteint le périmètre de la maison, le système met définitivement le réfrigérateur en mode luxe, c'est-à-dire que lorsque le modèle de réfrigérateur reçoit l'événement «CCE» sur son port '1' et avec VEC=100, il passe en phase «Augmenter» au maximum. Cette transition dans le cas a également été effectuée par la fonction de transition externe.

#### IV.4.5 Résultats de simulation durant 24 heures

Maintenant, nous comparons dans la figure III.9 les résultats de consommation de l'énergie des appareils électroménagers dans la maison, sur 24 heures, dans le cas du mécanisme appliqué, avec les résultats de la consommation d'énergie des mêmes appareils dans la maison sans utiliser ce mécanisme qu'il est basé sur la localisation de l'utilisateur.

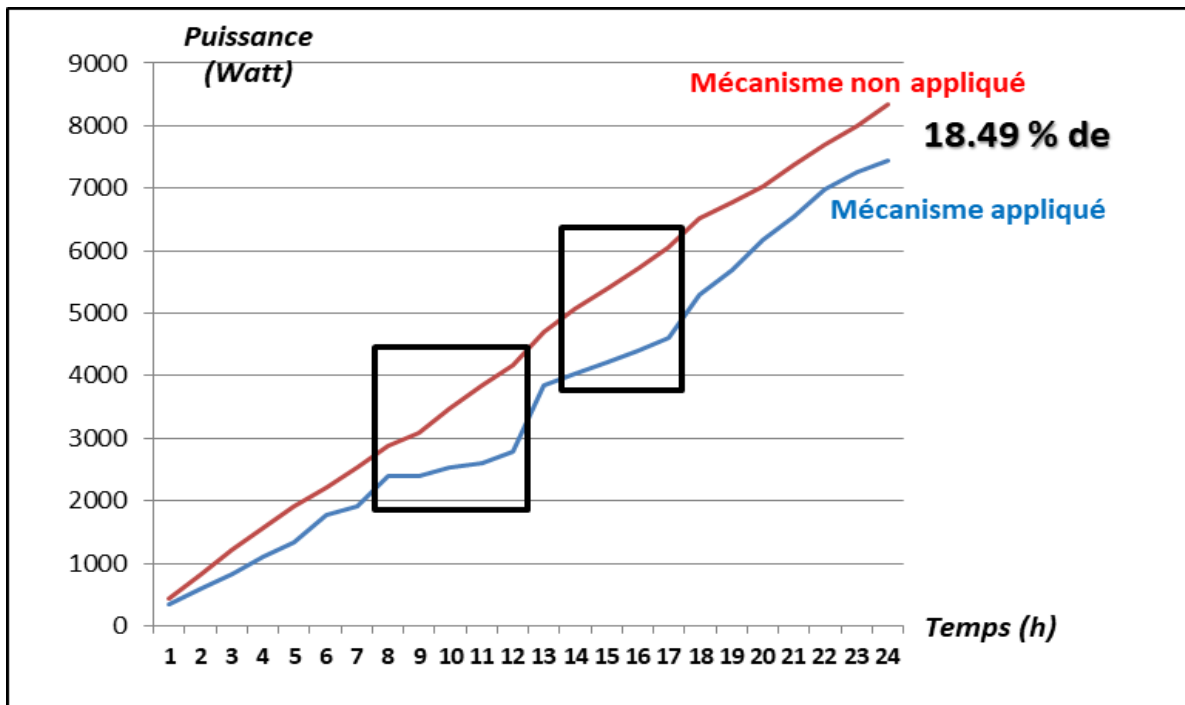


Figure III.9 : Gain énergétique quotidienne de la maison

À travers cette figure, on voit que la consommation énergétique effective de ce bâtiment, après application de ce mécanisme, est bien inférieure à celle des bâtiments qui ne le font pas ; En particulier, cet écart augmente rapidement pendant les moments où l'utilisateur atteint les limites du bâtiment (quand il quitte sa maison pour aller dehors), jusqu'à ce qu'en fin de journée on obtienne un gain énergétique intéressant : environ 18,5% d'énergie gagnée par rapport à une consommation non contrôlée.

Imaginez si les environs de l'application de ce mécanisme étaient une ville entière, alors nous obtiendrions sûrement un profit indescriptible de l'énergie et un luxe garanti pour tous les utilisateurs des bâtiments. Ce bénéfice est le résultat d'une optimisation intelligente de la gestion de la consommation d'énergie qui évite tout gaspillage qui se produirait en cas de consommation d'énergie incontrôlée. Nous pouvons dire que nous avons pu rationaliser l'utilisation énergétique d'un bâtiment intelligent de la manière la plus pratique en raison de sa clarté et de son efficacité.

#### IV.4.6 Résultats de simulation durant une semaine

En conclusion, afin de confirmer ces résultats et d'approfondir notre analyse, une troisième simulation a été réalisée sur une semaine entière. Par rapport à la consommation d'énergie réelle, la figure IV.10 montre la consommation d'énergie totale optimisée pour la maison, pendant une semaine entière.

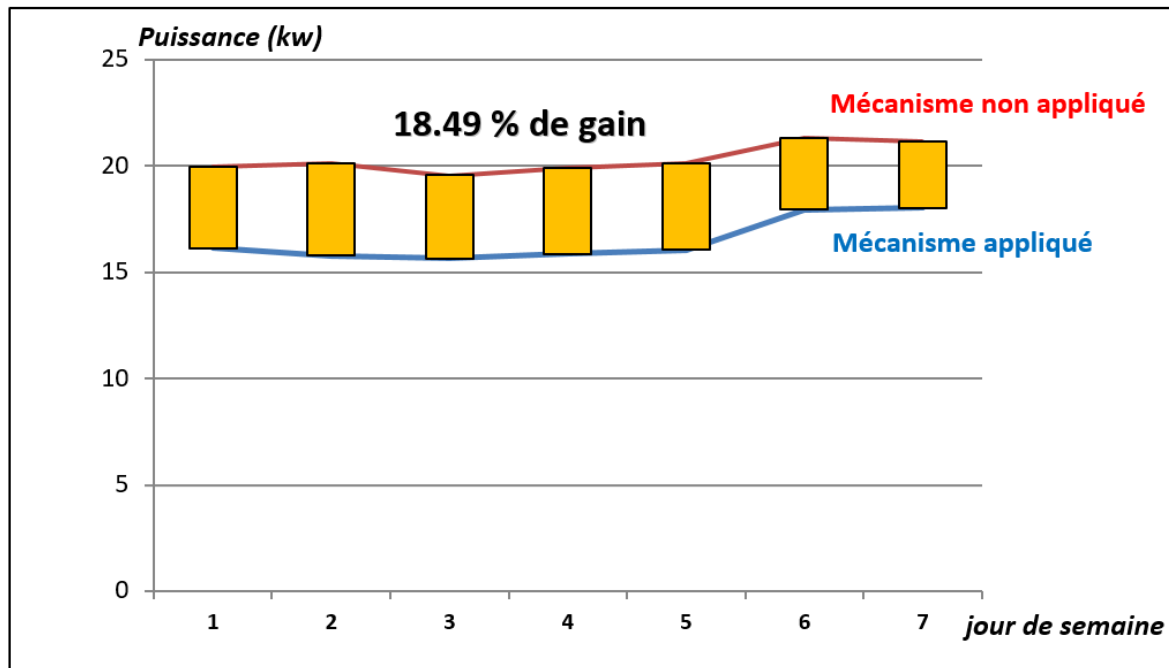


Figure IV.10 : Gain énergétique hebdomadaire de la maison

À travers les courbes présentées dans la figure, il apparaît après comparaison que nous avons gagné environ 18,76% de consommation totale d'énergie par semaine pour la maison après avoir appliqué l'approche d'optimisation basée sur la localisation de l'utilisateur.

Cela signifie qu'avec cette nouvelle approche, les utilisateurs généraux bénéficieront d'un style de vie luxueux sans aucun soin particulier ni tracas, et ils ne paieront que ce que les utilisateurs du mode économique paient.

Par ailleurs, nous notons et confirmons également en observant la rapidité de mise en œuvre que le secret de la performance de qualité de cette optimisation intelligente réside dans le respect minutieux des conceptions lors de l'exécution.

## **IV.5 Conclusion**

Afin d'étudier le comportement du système résultant de l'application de notre stratégie à son environnement approprié, nous avons choisi d'utiliser l'environnement de modélisation et de simulation JDEVS. Au début, Après avoir justifié notre choix de JDEVS, nous avons expliqué le rôle et le fonctionnement de chaque module contribuant à la composition de ce simulateur.

Dans un second lieu, nous avons implémenté tous nos modèles en utilisant JDEVS et nous avons effectué une comparaison de la consommation d'énergie résultante après une simulation durant 24 heures, avec l'estimation de la consommation quotidienne en trois modes de consommation, luxe, ordinaire et économique; Cette comparaison prouve que nous avons atteint le luxe des utilisateurs au moindre coût. Ensuite, nous avons contrôlé le comportement d'un équipement temporaire et le comportement d'un équipement permanent dans tous les cas d'emplacement de l'utilisateur afin de tester notre optimisation attendue prouvant l'exactitude de notre modèle.

À la fin, nous avons discuté les résultats de simulation de la consommation énergétique totale quotidienne et hebdomadaire, ce qui nous a montré qu'avec notre approche, nous gagnions beaucoup d'énergie sans compromettre le bien-être de l'utilisateur.

# **Conclusion générale**

## **Conclusion générale**

Le secteur du bâtiment est une source potentielle majeure d'économie d'énergie, notamment grâce à la rationalisation énergétique, qui a suscité l'intérêt de nombreux chercheurs à contribuer à son développement à travers de nombreuses et importantes propositions.

Dans ce travail, nous avons présenté une nouvelle approche pour optimiser la consommation d'énergie dans les bâtiments intelligents en se basant sur la localisation de l'utilisateur. Pour déterminer la localisation de l'utilisateur par rapport au bâtiment qu'il utilise, nous nous sommes appuyés sur un mécanisme rationnel et intelligent. Un chapitre entier est consacré à une définition détaillée de toutes les parties qui contribuent à la mise en œuvre de ce mécanisme, y compris le réseau de capteurs sans fil qui jouent le rôle du système d'automatisation du bâtiment, les équipements consommateurs d'énergie et la localisation GPS.

Pour concevoir les systèmes résultants, nous nous sommes appuyés sur un environnement compatible, composé d'un bâtiment, et l'utilisateur accompagné toujours de son smartphone. Cet utilisateur peut quitter son domicile pour aller travailler à l'extérieur, par exemple, puis rentrer chez lui. Nous avons pensé à choisir cet environnement parce que nous voulions qu'il soit une porte d'entrée clé pour créer des environnements plus vastes et plus complexes, comme une ville entière, que nous aspirons à trouver des solutions pour rendre plus intelligents. Nous avons pris en compte ce phénomène dans nos conceptions, depuis le début de l'explication de la scène générale jusqu'à la conception des systèmes résultants, jusqu'à l'ajustement des modes d'utilisation de l'énergie en fonction de la localisation de l'utilisateur, dans le but de rationaliser sa consommation d'énergie.

Afin de valider notre approche, nous avons modélisé tous les systèmes résultants selon le formalisme DEVS après avoir appliqué ce système à son environnement approprié tout en préservant tous les scénarios possibles. Nous justifions notre choix du DEVS par son caractère modulaire exploité dans cette étude où nous pouvons former des sous-modèles, qui nous permettent de développer des techniques de simulation de manière indépendante, puis de les intégrer jusqu'à ce que nous atteignons le modèle global. Cette division des systèmes agrégés facilite le processus de modélisation, quelle que soit la complexité des systèmes.

Après avoir implémenté tous nos modèles dans l'environnement de modélisation et de simulation JDEVS, nous avons adopté une approche systématique pour analyser les résultats de la simulation en sélectionnant des scénarios appropriés pour cela. Nous avons exploré les facteurs qui avaient le plus d'influence sur la mise en forme des scènes, les principaux acteurs, en particulier les modèles de consommation d'énergie, les emplacements potentiels de l'utilisateur et les événements critiques qui mènent à des changements de situations, et après avoir analysé les résultats de simulation dans ces situations, nous avons montré que nous obtenons des économies d'énergie significatives sans compromettre le bien-être des utilisateurs.

En tant que perspectives d'avenir, nous espérons trouver des solutions en cas de grand nombre de bâtiments, y compris le problème de la localisation de l'utilisateur par rapport à l'ensemble de ces bâtiments, que nous envisageons de résoudre en créant un algorithme qui compare en permanence la localisation de cet utilisateur avec toutes les coordonnées qui définissent chaque bâtiment séparément ; Cela se fera bien sûr après que toutes ces coordonnées auront été préalablement enregistrées. D'autre part, nous pouvons enregistrer un grand nombre d'utilisateurs pour se connecter à notre système via une base de données spécifique qui donne à chaque utilisateur un numéro d'identification unique pouvant être utilisé dans l'algorithme envisagé, et à partir de là, nous pouvons passer à l'optimisation de la gestion d'énergie pour toute une ville.

# Références

## Références

- [Abyan, 2019] A. Y. Abyaneh, V. Pourahmadi, A-H-G Foumani. CSI-based authentication: Extracting stable features using deep neural networks. *Trans. Emerging Tel. Tech*, 30(11), 1–11. 2019. [https://doi: 10.1002/ett.3795](https://doi.org/10.1002/ett.3795).
- [Aid, 2016] L. Aid, L. Zaoui, S. A. M. Mostefaoui. Using DEVS for modeling and simulation of ambient objects in intelligent buildings. *Journal of Ambient Intelligence and Humanized Computing*, 7(4): 579-592. 2016
- [Bandy, 2013] Bandyopadhyay, S. and Bhattacharyya, A., "Lightweight Internet protocols for web enablement of sensors using constrained gateway devices", International Conference on Networking and Communications (ICNC), vol., no., pp.334,340, 28-31 Jan. 2013.
- [Belha, 2020] A. Belhadi, Y. Djenouri, G. Srivastava, D. Djenouri, J. C. W. Lin, G. Fortino. Deep learning for pedestrian collective behavior analysis in smart cities: A model of group trajectory outlier detection. *Information Fusion*, 65: 13–20. 2021. <https://doi.org/10.1016/j.inffus.2020.08.003>.
- [Berna, 2002] F. Bernardi. "Conception de bibliothèques hiérarchisées de modèles réutilisables selon une approche orientée objet." Thèse de Doctorat, Université de Corse. 2002.
- [Bouke, 2013] Boukelkoul, S., Redjimi, M., 2013. Mapping between Petri nets and DEVS models, in: 2013 3<sup>rd</sup> International Conference on Information Technology and E-Services (ICITeS). Presented at the 2013 3rd International Conference on Information Technology and e-Services (ICITeS), pp. 1–6. doi:10.1109/ICITeS.2013.6624067.
- [Brand, 2000] J. Brandmeyer, H. Karimi. Coupling methodologies for environmental models. *Environmental Modelling and Software*, 15(5): 479-488. 2000.
- [Cassa, 1999] Cassandras, C. et S. Lafortune., Introduction to Discrete Event Systems, Kluwer Academic Publisher. 1999.
- [Cesco, 2020] J. Cesconetto, L. Augusto Silva, F. Bortoluzzi, M. Navarro-Cáceres, C. A. Zeferino, V. R. Q. Leithardt. PRIPRO—Privacy Profiles: User Profiling Management for Smart Environments. *Electronics*. 2020; 9: 1519.
- [Colit, 2011] Colitti, Walter, Kris Steenhaut, and Niccolò De Caro. "Integrating wireless sensor networks with the web", Sensor Networks Conference 2011.



[**Conne, 2021**] Connectwave. *Comment se compose un système IoT ?* Available online November 22 : <https://www.connectwave.fr/techno-appli-iot/iot/reseaux-et-infrastructures-iot/>

[**Decui, 2010**] J. Decuir, "Bluetooth 4.0: Low Energy", Presentation slides, 2010.

[**D'abre, 2003**] D'Abreu, M.C., Wainer, G., 2003. Models for continuous and hybrid system simulation, in: Simulation Conference, 2003. Proceedings of the 2003 Winter. Presented at the Simulation Conference, 2003. Proceedings of the 2003 Winter, pp. 641–649 Vol.1. doi:10.1109/WSC.2003.1261479

[**Erber, 1995**] erber, J. 1995. Les systèmes multi-agents. Vers une intelligence collective. InterEditions, Paris.

[**Filip, 2013**] Filippi, J.-B., Pialat, X., 2013. Assessment of FireFire/Meso-NH for wildland fire/atmosphere coupled simulation of the FireFlux experiment. Proc. Combust. Inst. 34, 2633–2640. doi:10.1016/j.proci.2012.07.022

[**Filip, 2004**] J-B Filippi, P. Bisgambiglia. JDEVS: an implementation of a DEVS based formal framework for environmental modeling. *Environmental Modelling & Software*, 19(3) : 261–274. 2004. <https://doi.org/10.1016/j.envsoft.2003.08.016>.

[**Fishm, 2001**] Fishman, G.S., 2001. Discrete-Event Simulation: Modeling, Programming, and Analysis. Springer Science & Business Media.

[**Flax, 1991**] B. M. Flax. Intelligent buildings. IEEE Communications Magazine, 29(4) : 24–27. 1991. <https://doi.org/10.1109/35.76555>

[**Jia, 2019**] Jia, M., Komeily, A., Wang, Y., & Srinivasan, R. S. (2019). Adopting Internet of Things for the development of smart buildings: A review of enabling technologies and applications. *Automation in Construction*, 101, 111–126.

[**Joche, 2018**] Jochen Wirtz, Paul G Patterson, Werner H Kunz, Thorsten Gruber, Vinh Nhat Lu, Stefanie Paluch, and Antje Martins. Brave new world : service robots in the frontline. *Journal of Service Management*, 2018.

[**Julio, 2013**] Julio Arauz and Tony Fynn-Cudjoe. Actuator quality in the internet of things. In 2013 IEEE International Conference on Sensing, Communications and Networking (SECON), pages 34–42. IEEE, 2013.

[**Lara, 2002**] Lara, J. de, Vangheluwe, H., 2002. Computer Aided Multi-paradigm Modelling to Process Petri-Nets and Statecharts, in: Proceedings of the First International Conference on

GraphTransformation, ICGT '02. Springer-Verlag, London, UK, UK, pp. 239–253.

[**Leemi, 2006**] Leemis, L.M., Park, S.K., 2006. Discrete-event Simulation: A First Course. Pearson Prentice Hall.

[**Luan, 2015**] T. H. Luan, L. Gao, Z. Li, Y. Xiang, L. Sun. Fog computing: Focusing on mobile users at the edge, Preprint, available online arXiv:1502.01815[cs.NI], (2015), 1–11.

[**Maato, 2021**] A. Maatoug, G. Belalem, S. Mahmoudi. A location-based fog computing optimization of energy management in smart buildings: DEVS modeling and design of connected objects. *Front. Comput. Sci.*, Just Accepted Manuscript. 2021. ISSN 2095-2228.

[**Maato, 2014**] A. Maatoug, G. Belalem. Conception and validation of smart building energy management system BEMS using the discrete event system specification DEVS. *Journal of Communications Software and Systems (JCOMSS)*, 10 (2) : 107–113. 2014.

[**Panch, 2019**].Panchalingam, R., & Chan, K. C. (2019). A state-of-the-art review on artificial intelligence for Smart Buildings. *Intelligent Buildings International*, 1–24.

[**Patel, 2016**] Patel, K. K., & Patel, S. M. (2016). Internet of things-IoT: definition, characteristics, architecture, enabling technologies, application & future challenges. *International journal of engineering science and computing*, 6 (5).

[**Ramad, 1987**] Ramadge, P. & Wonham, W. „Supervisory control of a class of discrete event processes“, *SIAM J. Control Optim.* 25(1), 206–230., [1987].

[**Ray, 2016**] Ray, P. P. (2016). A survey of IoT cloud platforms. *Future Computing and Informatics Journal*, 1 (1–2), 35–46.

[**Richa, 2006**] Richard Harper. *Inside the smart home*. Springer Science & Business Media, 2006.

[**Sampa, 1995**] Sampath, M., R. Sengupta, S. Lafortune, K. Sinnamohideen et D. Teneketzis., «Diagnosability of discrete event systems», *IEEE Transactions on Automatic Control*, vol.40, no 9, p. 1555–1575, [1995].

[**Sara, 2014**] Sara Amendola, Rossella Lodato, Sabina Manzari, Cecilia Occhiuzzi, and Gaetano Marrocco. Rfid technology for iot-based personal healthcare in smart spaces. *IEEE Internet of things journal*, 1(2) :144–152, 2014.

[**Sokol, 2010**] Sokolowski, J.A., Banks, C.M., 2010. Modeling and Simulation Fundamentals: Theoretical Underpinnings and Practical Domains. John Wiley & Sons.

**[Stojm, 2014]** I. Stojmenovic, S. Wen. *The fog computing paradigm: Scenarios and security issues*. In: *Federated Conference on Computer Science and Information Systems (FedCSIS)*, Warsaw, Poland, (Sep. 7–10, 2014), 1–8.

**[Vangh, 2002]** Vangheluwe, H., De Lara, J., Mosterman, P., 2002. An introduction to multi-paradigm modelling and simulation. Proc. AIS2002 Conf. AI Simul. Plan. High Auton. Syst. April 2002 Lisb. Port. F Barros N Giambiasi Eds 9–20.

**[Vasil, 2006]** Andra - Ioana VASILIU, Synthèse de contrôleurs des systèmes à événements discrets basée sur les réseaux de Pétri., [2006].

**[Wu, 2011]** G. Wu, et. al. "M2M: From Mobile to Embedded Internet", IEEE Communications Magazine ( Volume: 49, Issue: 4, April 2011 ).

**[Xiaol, 2012]** Xiaolin Jia, Quanyuan Feng, Taihua Fan, and Quanshui Lei. Rfid technology and its applications in internet of things (iot). In *2012 2nd international conference on consumer electronics, communications and networks (CECNet)*, pages 1282–1285. IEEE, 2012.

**[Zeigrl, 2000]** B. P. Zeigler, H. Praehofer, T. G. Kim. *Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems*. Academic press, San Diego (2000). ISBN: 0127784551.

**[Zeigl, 1976]** B. Zeigler. *Theory of Modeling and Simulation*, New York: *Wiley Interscience*, 1976.

**[Zeigl, 1993]** Zeigler, B.P., Vahie, S., 1993. Devs Formalism And Methodology: Unity Of Conception/Diversity Of Application, in: In *Proceedings of the 25th Winter Simulation Conference*. ACM Press, pp. 573–579.

**[Zimme, 2013]** Zimmer, S., Buchholz, M., Pflüger, D., 2013. *Modeling and Simulation: An Application-Oriented Introduction*, 2014 edition. ed. Springer.