

FINITE STATE MACHINE

---

# Contrôleur de maison intelligente Rapport

---

Salah-Eddine BARIOL  
ALAOUI

SI4 G2

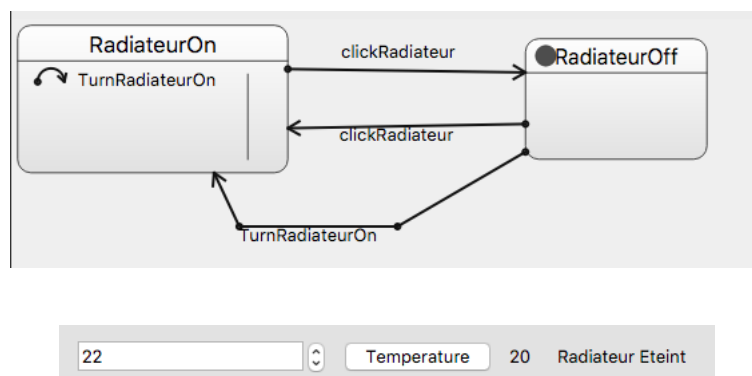
06 Janvier 2018

## Fonctionnalités principales

### 1.1 Programmation et régulation de chauffage

#### 1.1.1 Radiateur

Chaque pièce dispose d'un radiateur. L'utilisateur peut l'allumer ou l'éteindre en cliquant sur le bouton dédié. Lorsque le bouton est appuyé la machine l'évènement `clickRadiateur` est envoyé.



L'utilisateur sélectionne la température qu'il veut et clique sur le bouton pour allumer le radiateur.

À l'entrée de l'état **RadiateurOn** la statemachine envoie l'action `doAllumer` qui va allumer le régulateur avec comme température désirée la température de la `QSpinBox`. À l'entrée de l'état **RadiateurOff** la statemachine envoie l'action `turnOffTemp` qui va allumer également le régulateur avec comme température désirée la température de l'environnement (choisie arbitrairement à 20 degrés).

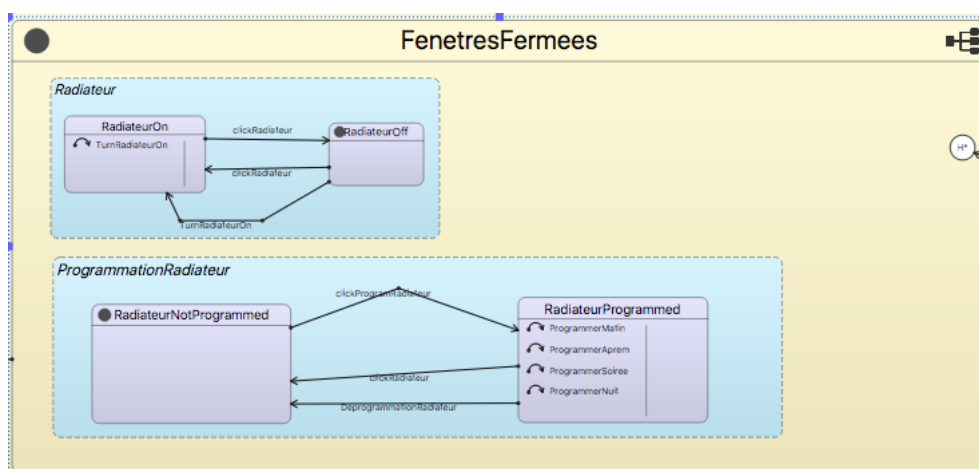
### 1.1.2 Régulateur

Le régulateur est ce qui permet de modifier la température. C'est les états du Radiateur qui décident si oui ou non il faut l'allumer en envoyant l'événement setOn. Ensuite en OnEntry de l'état Good on envoie l'événement CommencerRegulateur qui va comparer les températures et décider s'il faut envoyer l'évènement tooHot ou tooCold. Si les deux températures sont égaux l'évènement setOff est envoyé. En OnEntry de l'état notGood on envoie 1 seconde après l'évènement check.

### 1.1.3 Programmation du Radiateur

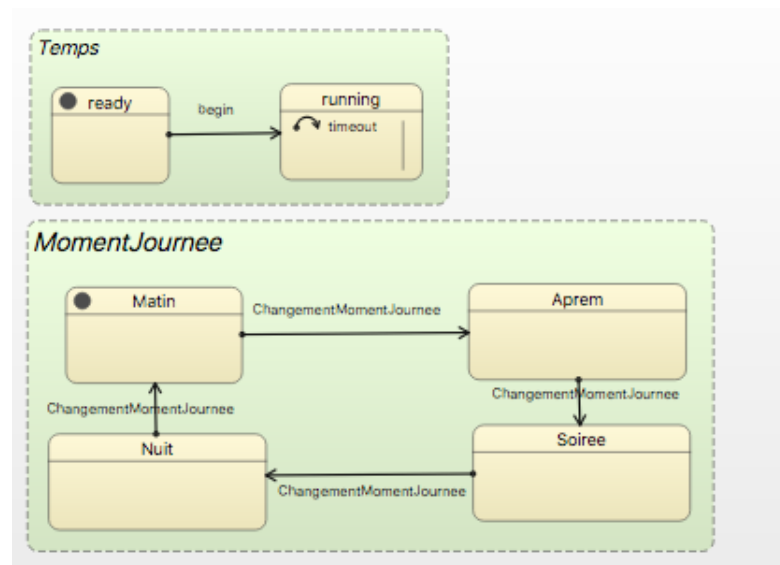
Il est possible de programmer le radiateur dans chaque pièce pour qu'il s'allume à des températures données.

L'utilisateur entre les températures qu'il veut pour chaque moment de la journée et clique sur "Programmer Température".



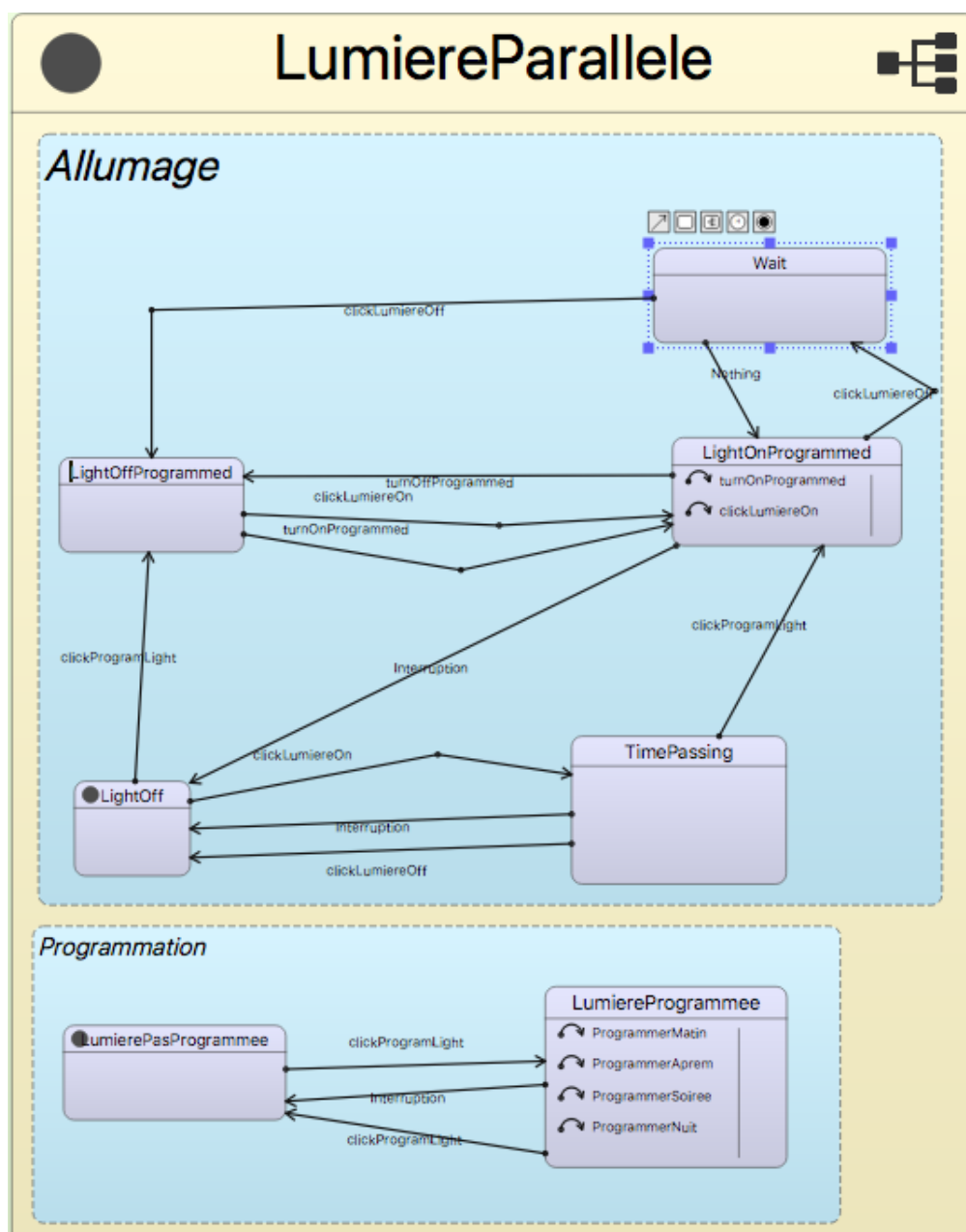
Lors d'un clique "Programmer Température" l'évènement "clickProgramRadiateur" est envoyé, on entre dans l'état "RadiateurProgrammed".

Ensuite on a une statemachine GestionAppartement qui contient l'état MomentJournée et l'état Temps en parallèle. En OnEntry de l'état running on envoie l'évènement timeout qui incrémente de 1 le temps. Pour simuler j'ai dit quand lorsque le temps modulo 10 = 0 on envoie l'évènement ChangementMomentJournée. Ensuite pour chaque état de MomentJournée, le matin par exemple on envoie l'évènement doDisplayMatin. Qui va envoyer pour chaque pièce l'évènement "ProgrammerMatin". Pour ensuite allumer le radiateur selon le programme de la matinée.



## 1.2 Gestion des lumières

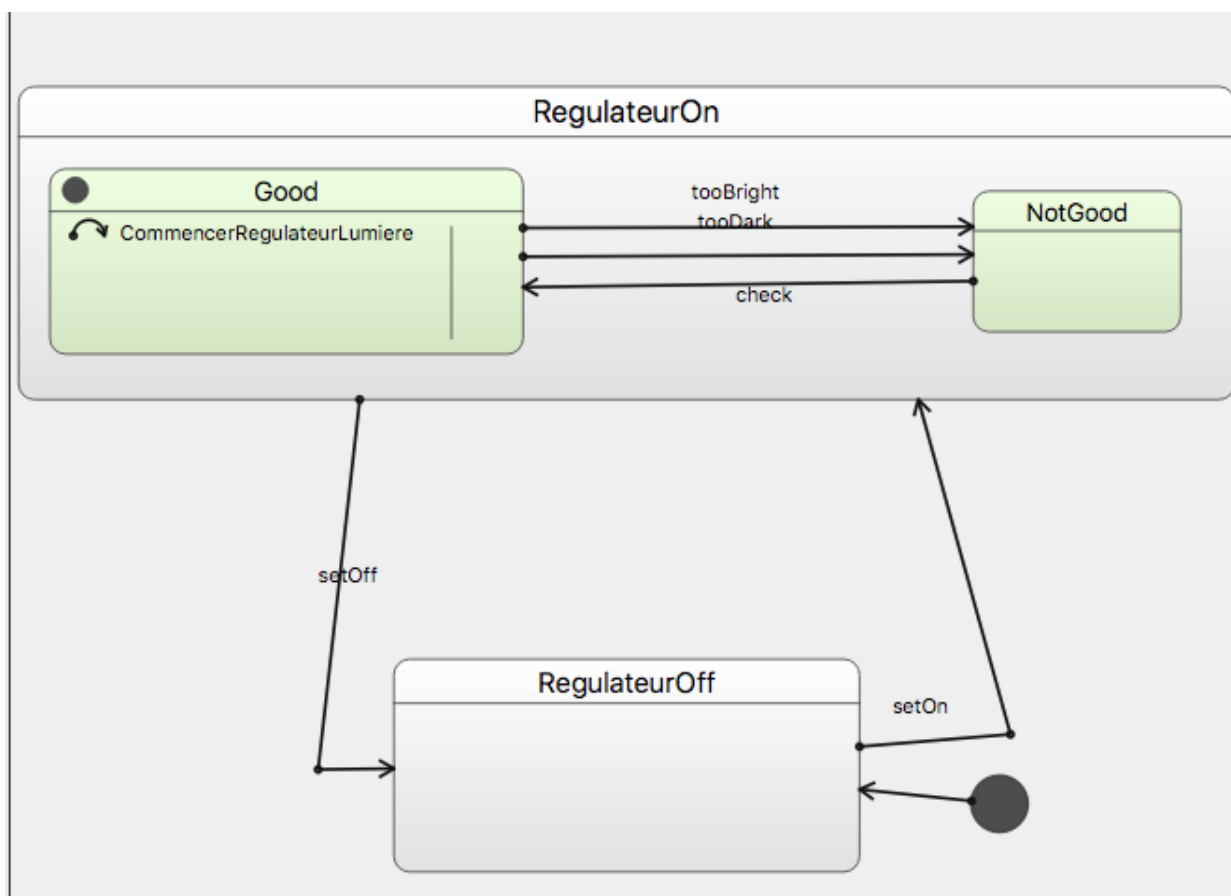
La gestion des lumières possède deux états en parallèles. L'allumage qui allume ou éteint le régulateur des lumières et la programmation des lumières qui dirige l'allumage en fonction si l'utilisateur a programmé la lumière.



En raison de l'extention du maintiens de luminosité minimum, j'ai utilisé deux boutons. Un bouton pour allumer et un autre pour éteindre. Le premier envoie l'évènement `clickLumiereOn` et le dernier `clickLumiereOff`. L'évènement `TurnOnProgrammed` est l'évènement envoyé par la programmation des lumières pour allumer les lumières. Lors d'un clique sur le bouton `Interruption` dans la centrale domotique toutes les pièces reçoivent l'évènement `Interruption` qui va éteindre les lumières et les déprogrammer.

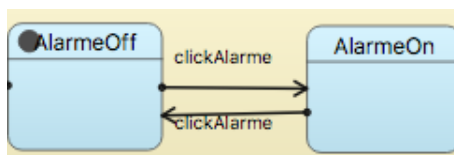
### 1.2.1 Régulateur de lumières

Le régulateur des lumières fonctionne de la même façon que celui du radiateur.



## 1.3 Alarme

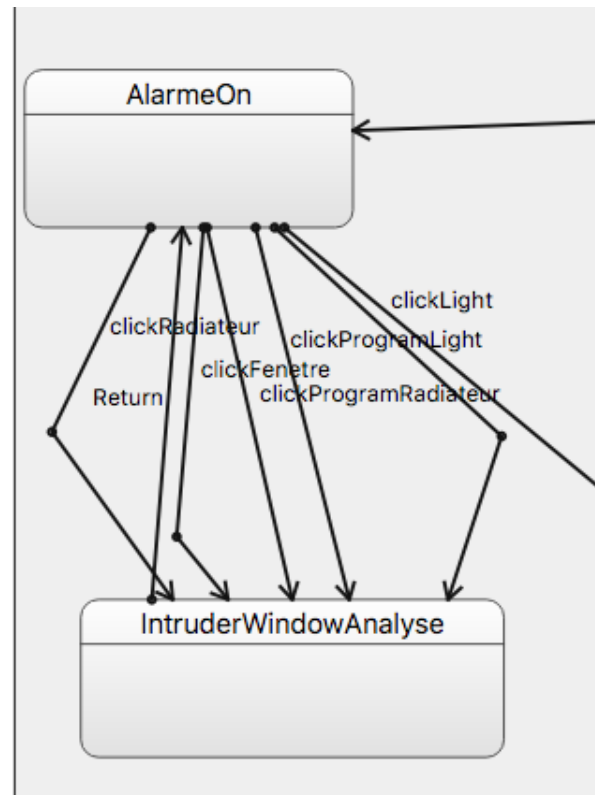
J'ai fait le choix de dire qu'on ne peut activer l'alarme que dans tout l'appartement, ce n'est pas possible de paramétrer l'alarme par pièce. Dans la centrale domotique l'utilisateur peut éteindre ou allumer l'alarme, un évènement **clickAlarme** est envoyé.



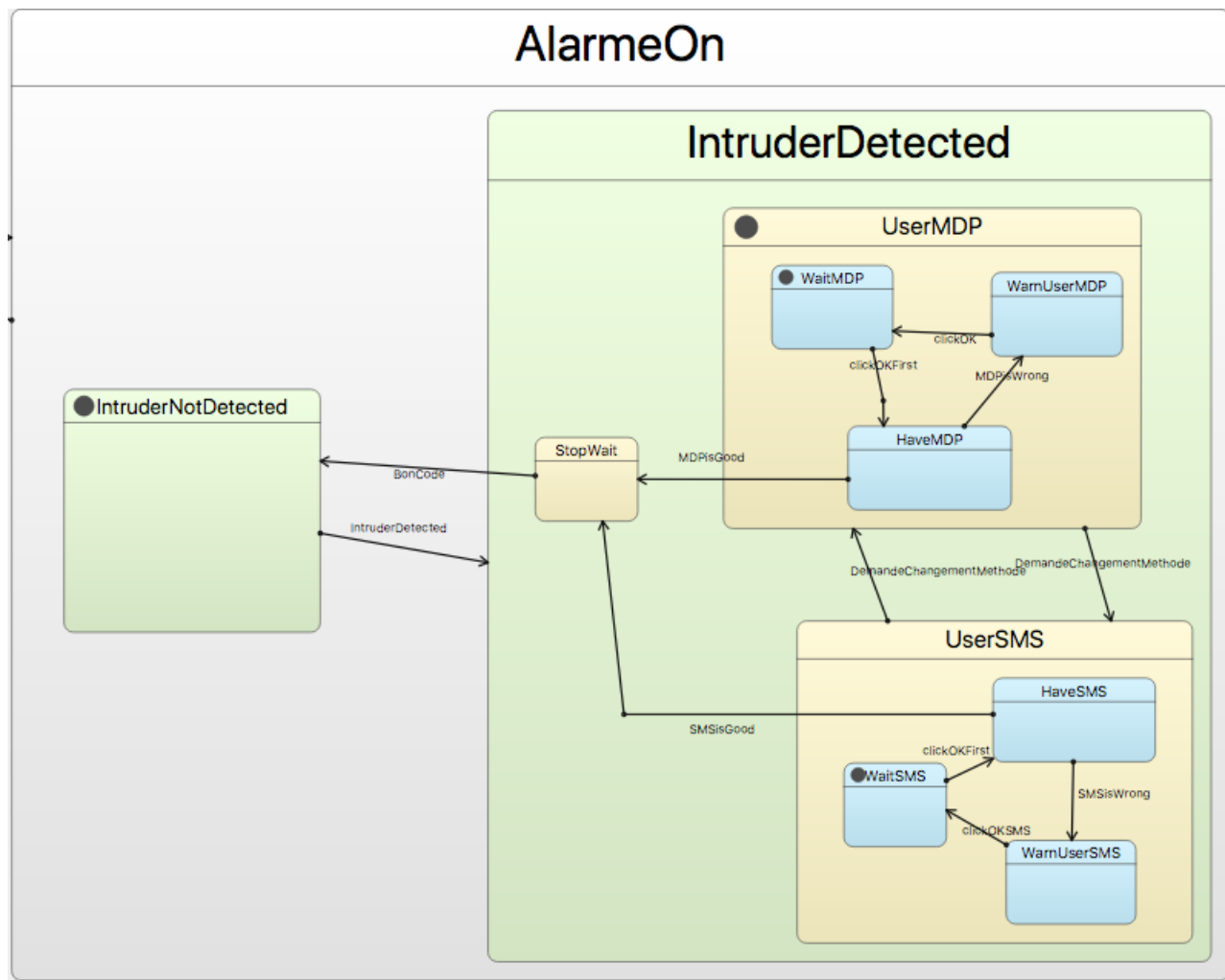
L'alarme est présente à plusieurs étages. Premièrement la centrale domotique qui contrôle tout l'appartement quand l'alarme est allumée elle allume l'alarme dans toutes les pièces.

Chaque pièce a ensuite sa propre alarme qui entre dans l'état AlarmeOn. La pièce entre également dans l'état AlarmeOn.

Dans la pièce on a :



Dans la statemachine de l'alarme on a :



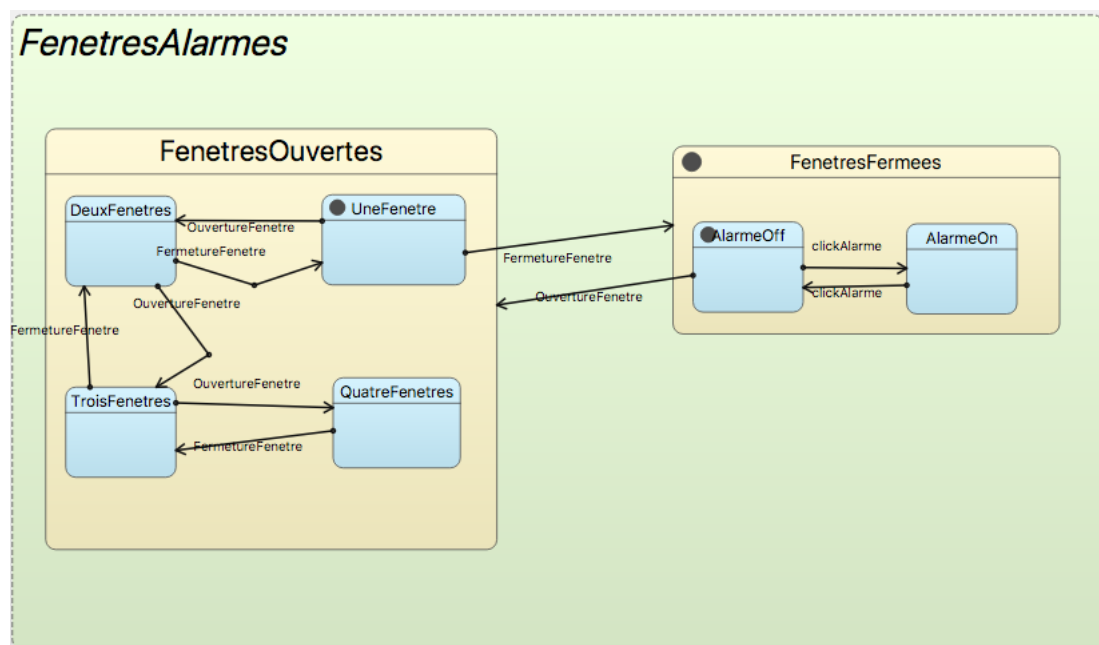
Si dans une pièce on a par exemple l'événement **clickLight** qui est produit alors on entre dans l'état **IntruderWindowAnalyse** qui va envoyer l'événement **IntruderDetected** à l'alarme. Dans l'alarme on entre dans l'état **IntruderDetected**. Une fenêtre apparaît pour que l'utilisateur entre le mot de passe où demande à recevoir un sms. Si le mot de passe ou le sms est entré on retourne dans l'état **IntruderNotDetected**, qui envoie l'événement **Return** à la pièce. Pour revenir dans l'état **AlarmeOn**.



## 2.1 Détection des fenêtres ouvertes

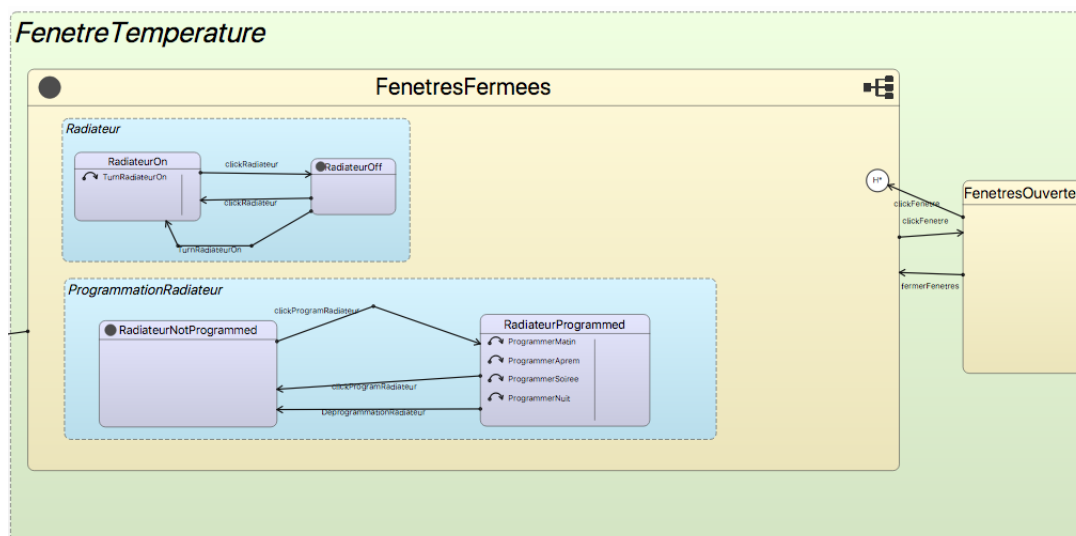
### 2.1.1 Centrale domotique

Dans la centrale domotique les fenêtres sont liées à l'alarme. On ne peut activer l'alarme que dans l'état FenetresFermees et on ne peut entrer dans l'état FenetresOuvertes que dans à partir de l'état AlarmeOff.



### 2.1.2 Pièce

Dans chaque pièce les fenêtres sont liées à la température.



On ne peut allumer le radiateur que lorsque les fenêtres sont fermées. Si on ouvre le radiateur s'éteint. Si on ferme les fenêtres on retourne dans l'état précédent et le radiateur se remet à fonctionner.

Pour toutes les pièces si on ouvre la fenêtre on lance l'événement OuvertureFenetre dans la centrale domotique.

## 2.2 Maintiens de luminosité minimum

Lorsque l'utilisateur programme les lumières d'une pièce il peut aussi choisir l'intensité minimum.

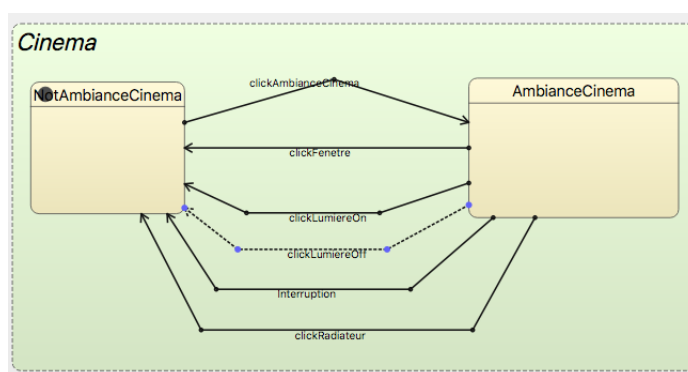
Programmation des lumières			
	Allumage (1/0)	Durée	
Matin	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="20"/>
Aprém	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="30"/>
Soirée	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="40"/>
Nuit	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="60"/>
<input type="button" value="Quitter programmation"/>			

Quand l'utilisateur clique sur Programmer l'évènement clickProgramLight est lancé, on entre dans LightOnProgrammed ou LightOffProgrammed selon que la lumière était allumée ou éteinte à la base. Si l'utilisateur allume manuellement l'évènement clickLumiereOn est envoyé. Cependant si l'utilisateur souhaite éteindre la lumière l'évènement clickLumiereOff est envoyé, on entre dans un état Wait qui attends. Si rien ne se passe

au bout de 3 secondes l'évènement Nothing est envoyé et on retourne dans l'état LightOnProgrammed si avant ces trois secondes l'utilisateur reclique l'évènement clickLumiereOff est de nouveau envoyé et on va dans l'état LightOffProgrammed qui va éteindre la lumière.

## 2.3 Home Cinema

Nous avons dans pièce nous avons une gestion du cinéma :



Si l'utilisateur clique sur le bouton Home Cinema alors l'évènement clickAmbianceCinema est envoyé. Dans le OnEntry de AmbianceCinema, on envoie plusieurs évènement. Dans l'ordre : on envoie l'évènement fermerFenetres, on sélectionne 25 degrés dans la radiateur et on envoie l'évènement TurnRadiateurOn, on envoie l'évènement clickLumiereOn. Cette façon de faire permet que les autres composants de la pièce reste modifiable par eux-mêmes.

## 2.4 Service personnalisé

Je n'ai pas fait cette extension.

Les statemachines permettent vraiment de se simplifier la vie