Road Runner

Road Runner

Conception du logiciel

Cuiseur de riz

Version 1.3

Mathieu Guérin
Martin Grenier
Israël Halle
René-Alexandre Giroux

Date (03-03-13)

Le document de conception du logiciel (Software Design Document, SDD) est une description de la conception qui permet d'orienter le développement vers l'architecture du logiciel.

Dropbox\Gabarits\GABARIT SDD.docx

Historique des révisions

Date (jj-mm-aaaa)	Version	Description	Auteur
27-02-2013	1.0	Création du document	René-Alexandre Giroux
27-02-2013	1.1	Contribution au document	Martin Guérin, Mathieu Grenier, René-Alexandre Giroux, Israël Halle
02-03-2013	1.2	Ajout des diagrammes de conception	Israël Halle
03-03-2013	1.3	Correction du document après revues	Israël Hallé

Abréviations/acronymes

Abré./Acro.	Définition
SRS	Software requirement spécifications
ISO	Organisation internationale de normalisation
CU[X]	Cas d'utilisation [numéro de]

Table des matières

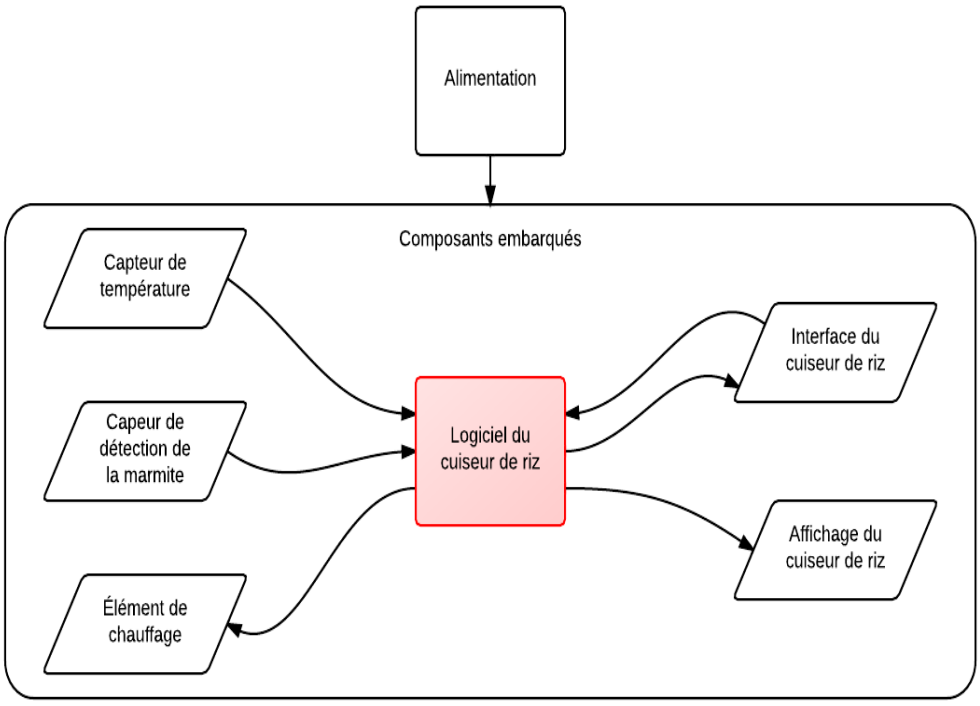
1. Introduction

1.1 Objectif

Le but du présent document est de fournir une description de la conception et de l'architecture logicielle en se basant sur les exigences fournies dans le document des spécifications du logiciel (SRS). Afin d'illustrer ces interactions, ce document contient la conception détaillée des éléments du logiciel ainsi que différentes vues représentant l'architecture logicielle.

1.2 Portée

Notre projet est limité au composant logiciel dans le microprocesseur du cuiseur de riz. Une équipe externe s'occupe de concevoir et développer les composantes électroniques du cuiseur de riz. Ces composantes fournissent les entrées et sorties montrées dans le graphique ci-dessous que notre logiciel utilisera afin de contrôler la température de cuiseur de riz selon les entrées utilisateurs.



1.3 Références

1.3.1 Références

Énoncé des travaux de la société Acme, Version 1.3, 2013-03-01,
Acme Enonce des travaux Ajout 3.pdf

Road Runner - GABARIT - Document d'architecture (SDD), 2013-03-01,
https://cours.etsmtl.ca/log330/private/Travaux/TP_2_Plan_du_projet/Road%20Runner%20GABARIT%20SDD.docx

1.3.2 Références normatives

ISO/CEI TR 29110-5-1-2, 2011-05-15, Première édition
https://cours.etsmtl.ca/log330/private/Travaux/TP_2_Plan_du_projet/ISO_29110-5-1-2_2011%28F%29-1.pdf

2. Parties prenantes de la conception et leurs préoccupations

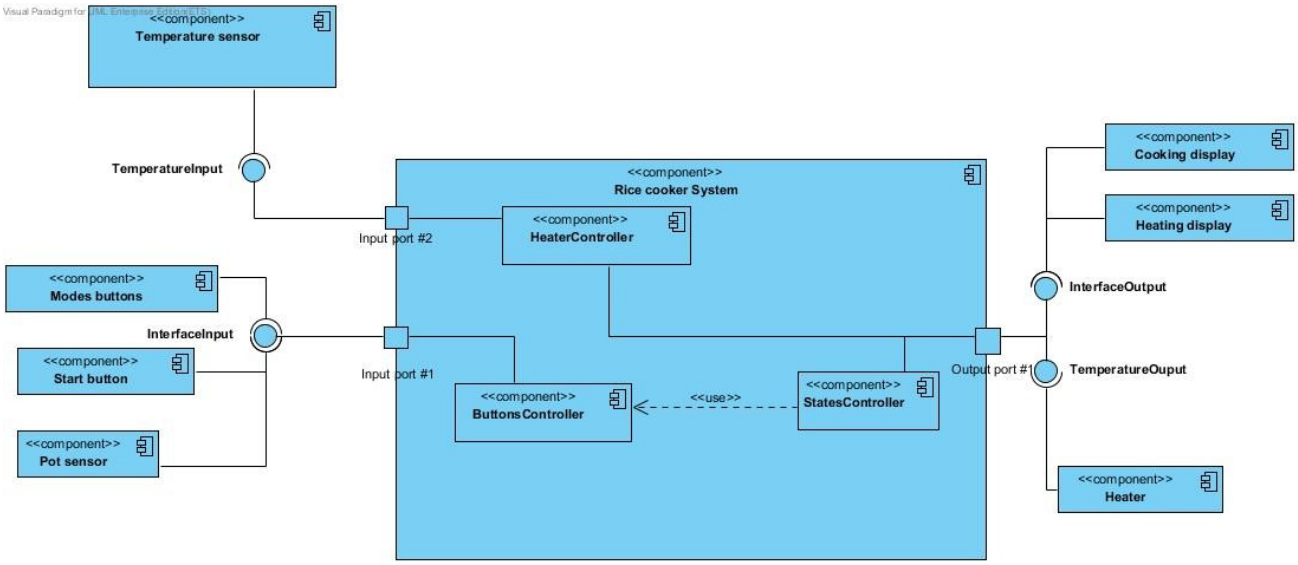
2.1 Parties prenantes de la conception et leurs préoccupations

Client	Que le cuiseur de riz soit fait à temps et fonctionnel pour la vente.
Architecte	Que l'architecture logicielle soit efficace et respecte les exigences.
Gestionnaire du projet	Que la conception soit réalisée dans les temps alloués en respectant les exigences.
Mainteneur	Que le logiciel soit maintenable et adaptable tout au long de sa durée de vie.
Utilisateur	Que le cuiseur de riz fonctionne.

3. Architecture logicielle

3.1 Vue d'architecture

3.1.1 Vue d'ensemble



Ce diagramme montre les ports d'entrée et de sortie et les composants qui les utilisent. Les ports d'entrée sont à gauche et les ports de sortie à droite.

3.1.2 Contraintes de conception qui s'applique à cette vue

CC1	Le logiciel doit être conçu pour fonctionner dans un environnement embarqué.
CC2	Le logiciel ne doit pas prendre trop de place mémoire dans le cuiseur de riz.

3.1.3 Exigences et préoccupations de conception

EC1	Le cuiseur offre 3 modes de cuisson.
EC2	Le cuiseur s'arrête automatiquement à la fin d'un cycle de cuisson.
EC3	Le cuiseur permet de trempage du riz pendant une heure.
EC4	Le cuiseur permet de conserver le riz au chaud pendant 4 heures.

EC5	Le cuiseur affiche qu'il est en opération.
EC6	Le cuiseur génère un signal sonore lorsqu'un cycle de cuisson se termine.
EC7	Le logiciel doit s'exécuter sans faille pour permettre d'effectuer des cuissons dans les trois modes de cuisson proposés.
EC8	Le logiciel doit être assez précis pour ne pas dépasser une marge d'erreur de 10 secondes lors de cycle de cuisson.
EC9	Le logiciel ne doit pas permettre des températures supérieures à 140°C.
EC10	Le logiciel doit permettre l'ajout de nouveau mode de cuisson.

3.1.4 Description des éléments de la vue et leurs interfaces

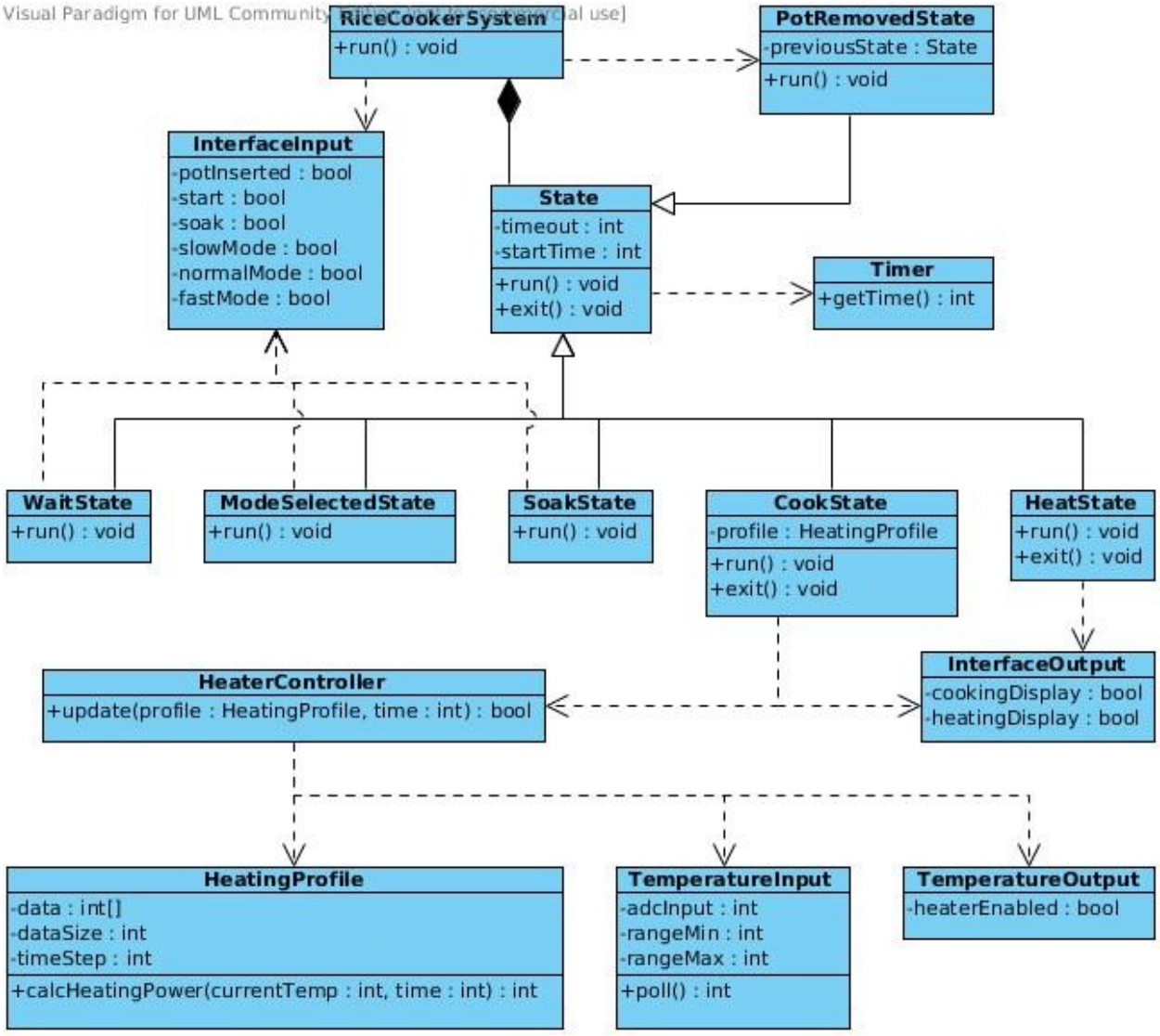
Le composant « StatesController » regroupe tous les états du système et leur logique. Le composant « ButtonsController » gère le déclenchement des événements lorsqu'un bouton est sur l'interface du cuiseur de riz est cliqué. Finalement, le composant « HeaterController » gère la puissance nécessaire à l'élément chauffant pour que le cuiseur de riz puisse suivre une courbe de température. Les ports d'entrées sont les boutons disponibles sur l'interface du cuiseur de riz et les capteurs de température et du pot. Les sorties sont les diodes sur l'interface du cuiseur de riz ainsi que l'élément chauffant.

3.1.5 Raisonnement

La plus grande partie du logiciel du cuiseur de riz est la gestion des différents états du cuiseur et leur transition. Ainsi, un composant est responsable de la gestion de celles-ci. Le contrôle de la température a été abstrait du composant de gestion des états. Le contrôle de la température est assez compliqué et demande une utilisation précise de formule thermodynamique pour suivre une courbe de température. Pour ces raisons, un composant y est dédié. Finalement, un dernier composant est responsable de surveiller l'état des entrées pour détecter le changement d'état d'un bouton et notifier la machine à état.

4 Conception détaillée

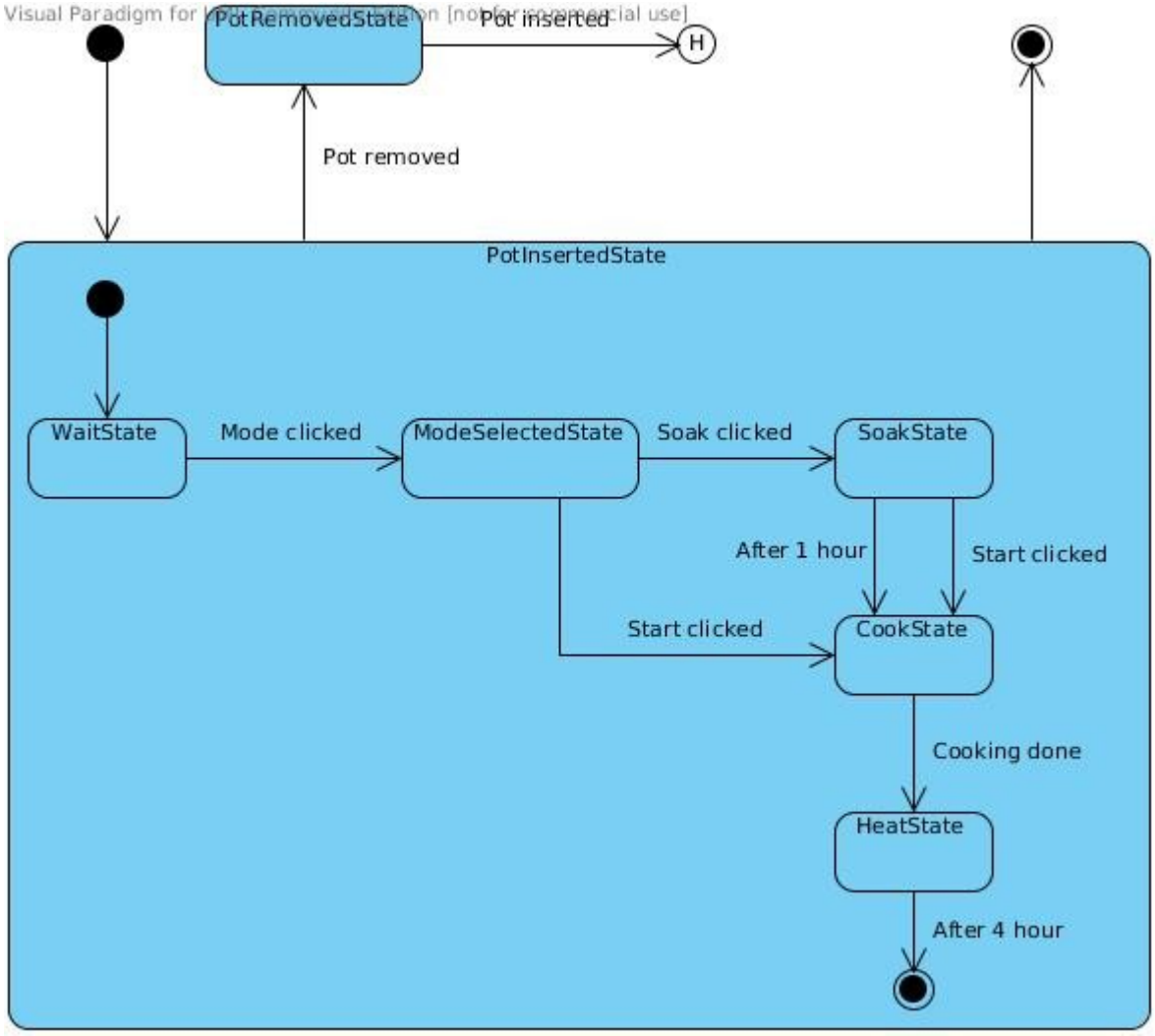
4.1 Vue structurelle



Ce diagramme de classe montre les différentes classes de notre système ainsi que leurs interactions avec les autres classes. La classe « RiceCookerSystem » est la classe centrale responsable de gérer la machine à état du logiciel. Celle-ci détecte lorsque la marmite est retirée afin de mettre la machine à état du système en pause

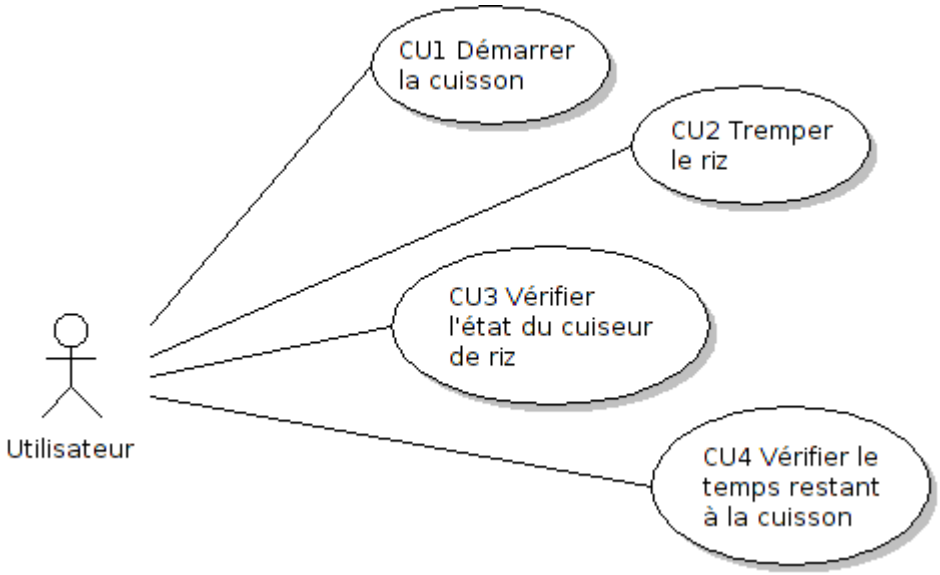
en attendant que celle-ci est remise. Les classes héritant de « State » définissent les différents états possibles du système. La plus importante est l'état « CookState ». Celle-ci est responsable de contrôler la cuisson du riz. La classe « HeaterController » est dédiée à la gestion de l'élément chauffant. La puissance nécessaire est calculée par « HeatingProfile » à partir de la température lue et du temps de cuisson écoulé. Les attributs de « HeatingProfile » contiennent les données nécessaires pour reproduire une des courbes de température.

4.2 Vue comportementale



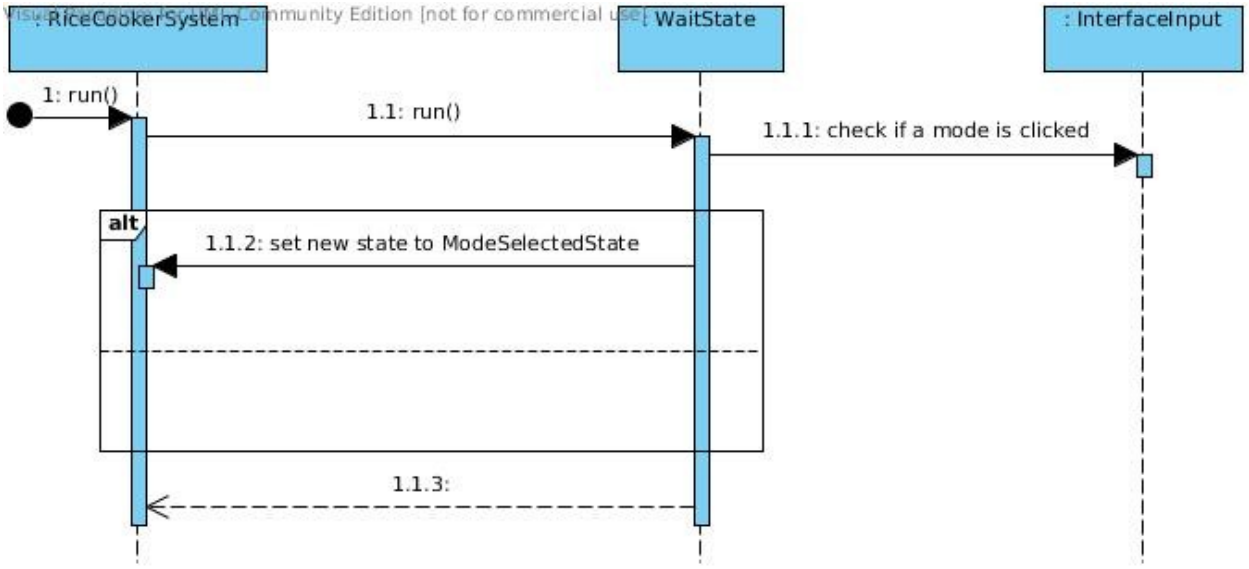
Ce diagramme montre les états et transitions possibles du logiciel du cuiseur de riz. Le premier état « WaitState » attend que l'utilisateur sélectionne un mode. Ensuite, « ModeSelectedState » attend la commande de départ ou de trempage.

« SoakState » laisse le riz tremper une heure ou attend que l'utilisateur appuie sur le bouton « Start ». « CookState » est l'état où le cuiseur de riz suit la courbe de température du mode précédemment sélectionner afin de cuire le riz. Lorsque la courbe de cuisson est terminée, le cuiseur passe au dernier état « HeatState » qui garde le riz chaud pendant 4 heures. Si au cours de cette machine à état la marmite est retirée, le système entre dans l'état « PotRemovedState ». Cet état attend simplement que la marmite soit replacée avant de retourner au dernier état.

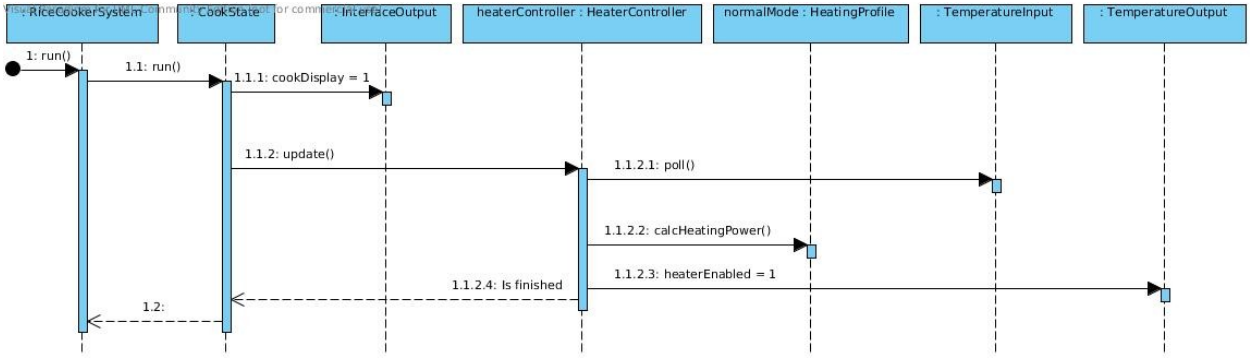


Ce diagramme de cas d'utilisation montre les cas d'utilisation du cuiseur de riz. CU1 est possible grâce en passant par l'état « SoakState » dans le diagramme de classe. Les autres cas d'utilisation sont réalisés par un parcours normal de la machine à état.

4.3 Autres vues pertinentes



Ce diagramme de séquence montre à quoi ressemble la transition d'un état à une autre. « RiceCookerSystem » est responsable d'exécuter la méthode « run » de l'état courant. Dans ce cas-ci, « WaitState » teste si un bouton d'un mode de cuisson est appuyé. Dans le cas échéant, l'état courant du système est changé pour l'état « ModeSelectedState ». Le même genre de séquence est réutilisé par tous les états en remplaçant le signal 1.1.1 par le traitement produit par l'état.



Ce diagramme montre la séquence des appels effectués lorsque le système est en état de cuisson. « CookState » débute en allumant la diode sur l'interface. Par la suite, elle met à jour le contrôleur de température. Celui-ci lit la dernière valeur de température du cuiseur de riz et calcule la puissance nécessaire à l'élément chauffant pour suivre la courbe de température du mode sélectionné. Finalement, le contrôleur envoie le message au port contrôlant l'élément chauffant.