

# Gestion du chauffage d'une maison

Eric Cariou et Wilfrid Lefer

Projet commun Systèmes Distribués/Système d'Exploitation

UPPA – 2011/12

Ce document décrit le sujet du projet commun entre l'UE de système d'exploitation et l'UE de systèmes distribués. Il est à faire en binôme. Il sera rendu un rapport par partie, suivi d'une évaluation sur machine.

## 1 Description générale de l'application à développer

On désire simuler le fonctionnement d'un système de gestion du chauffage dans une maison. Une maison est constituée de plusieurs pièces et chacune possède un appareil de chauffage. Chaque pièce possède également un thermomètre qui mesure la température de la pièce. La température d'une pièce évolue à la fois selon la température extérieure et le niveau de chauffage de la pièce.

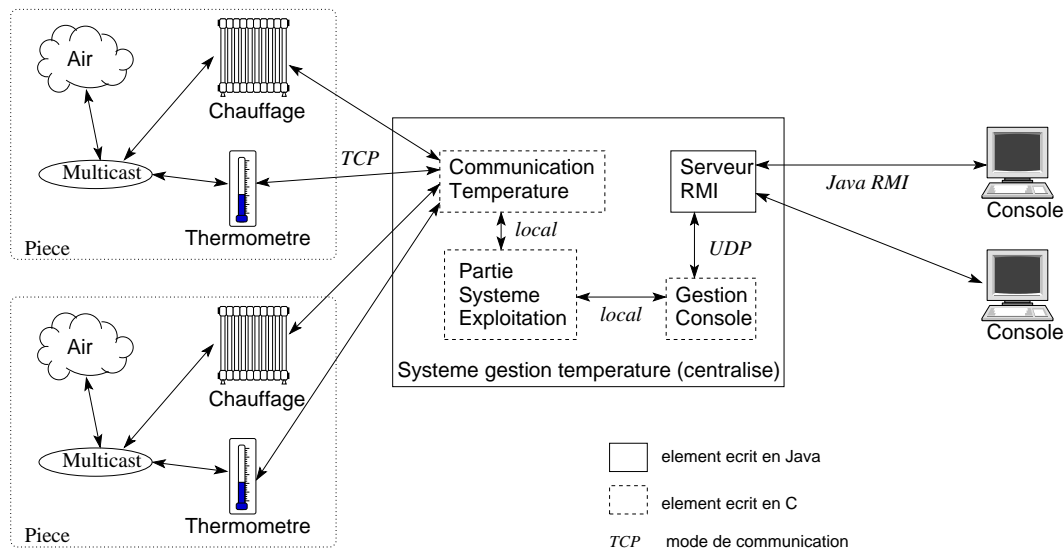


FIG. 1 – Architecture générale de l'application

L'application est constituée d'un ensemble d'éléments étant distribués, comme représenté sur la figure 1 :

- Des pièces qui contiennent chacune un thermomètre et un chauffage
- Un système de gestion de température qui coordonne les divers éléments entre eux

- Une ou plusieurs consoles permettant d’afficher les températures de toutes les pièces et de modifier le niveau de chauffage dans une pièce

## 2 Description des éléments formant l’application

### 2.1 Une pièce

Une pièce n’est pas un élément concret mais formé d’un ensemble de 3 éléments indépendants :

- Un chauffage qui permet de chauffer l’air de la pièce
- Un thermomètre qui mesure l’air de la pièce
- L’air de la pièce qui est à une certaine température

L’air correspond au programme Java `Air.java` qui vous est fourni et qui simule la variation de la température de l’air en fonction du niveau de chauffage et de la température extérieure. Les éléments thermomètre et chauffage sont à implémenter (en Java également).

La communication entre ces 3 éléments est à réaliser via du multicast UDP/IP. Il y aura donc un groupe multicast par pièce. La classe `MessageTemperature.java` (fournie également) correspond au type de message qui transitera entre les éléments du groupe multicast. Il permet d’envoyer deux types d’informations :

- La température courante de l’air de la pièce (information envoyée par l’élément Air)
- Un réchauffement de l’air de la pièce (information envoyée par l’élément chauffage) en précisant le niveau de puissance du chauffage (allant de 0 (arrêté) à 5 pour le chauffage le plus fort).

Au lancement de l’élément Air d’une pièce, on précisera l’identification du groupe multicast à utiliser ainsi que le nom de la pièce.

La communication entre un thermomètre ou un chauffage et le système de gestion des températures se fera en TCP.

### 2.2 Une console

Une console est l’interface permettant de gérer le chauffage. Il y a deux types de consoles :

- Un afficheur qui affiche en permanence la température courante de chaque pièce et l’état de chaque chauffage (arrêté ou allumé en précisant la puissance utilisée).
- Un contrôleur qui permet de modifier l’état de chaque chauffage.

Le contrôle de la température d’une pièce pourra se faire de 2 façons :

- Manuellement : on change à la main le niveau de chauffage de chaque pièce
- Automatiquement : on programme une température souhaitée pour une pièce et automatiquement le chauffage y sera allumé ou éteint selon la température courante de la pièce

La console communique avec le système de gestion de température via le middleware Java RMI.

### 2.3 Le système de gestion de température

C’est le cœur de contrôle du système de chauffage de la maison, tout passe par lui. Il devra être multiprocessus. Il sera composé de quatre processus : un processus

qui calculera l'ordonnancement des ordres aux radiateurs, un autre qui se chargera de la communication, un troisième qui gèrera la sécurité et l'intégrité du système, et enfin un dernier qui se chargera de la communication avec l'utilisateur. Les processus partageront un espace de mémoire partagée. Un cycle de calcul dure 5 secondes (ce temps n'est pas réaliste mais doit permettre une meilleure observation du fonctionnement de l'application ; ce temps doit être par ailleurs paramétrable au lancement de l'application).

### 2.3.1 Le processus Ordonnanceur

Le travail de ce processus est double. Il doit d'une part déterminer à chaque cycle si l'ordre à donner à chaque pièce doit être modifié ou non, cela en fonction des informations reçues de la part des consoles et des capteurs. D'autre part, il a pour rôle de déterminer l'ordre dans lequel les actions sur les éléments de la pièce seront envoyées. Pour cela, il gère une file multi-niveaux à retour représentant les 3 niveaux de criticité. En plus de cela, il doit gérer le temps d'attente de chaque pièce. Si une pièce n'est pas traitée depuis longtemps, sa priorité augmente. Cette liste doit être établie à chaque cycle.

### 2.3.2 Le processus Communication Température

Ce processus reçoit les informations provenant des pièces et les écrit en SHM. Il émet également les ordres du contrôleur à l'intention des pièces concernées.

### 2.3.3 Le processus Sécurité

Ce processus surveille le fonctionnement de l'application. Pour cela, il s'assure qu'à chaque cycle les processus ont tous terminé ce qu'ils avaient à faire. Pour cela, il surveille chaque processus avec un *chien de garde* (un cycle type pour un processus correspond à une suite d'actions : `chien de garde = 1 ; fonctionnel du processus ; chien de garde = 2 ; armerTimer de t secondes ; endormissement`). Si le processus de sécurité détecte trois débordements consécutifs, il arrête l'application après en avoir averti le contrôleur et avoir marqué le problème dans un fichier log.

### 2.3.4 Communication avec les éléments distribués

Le système de gestion communique avec deux ensembles différents d'éléments. D'un côté, il communique avec les consoles pour leur envoyer les informations venant des pièces et pour recevoir les ordres de commande sur les chauffages. Pour cela, il contient un objet Java particulier (nommé Serveur RMI sur la figure). Cet objet doit communiquer avec le reste du système de gestion, qui lui est écrit en C. Comme il n'existe pas de manière simple de faire communiquer localement un élément en Java et un élément en C, on passera par une communication en UDP. L'objet Java nommé Serveur RMI communique donc via des sockets TCP avec un élément C nommé Gestion Console qui fera alors l'interface avec le reste du système de gestion.

De l'autre côté, les chauffages et thermomètres (écrits en Java) communiquent via des sockets UDP standards avec l'élément du système nommé Communication Température qui lui est écrit en C.

### 3 Travail à rendre et évaluation

Vous rendrez pour le **XXXXXXX** un rapport écrit (format PDF) pour chacune des parties (systèmes distribués et système d'exploitation). Une évaluation sur machine aura lieu pour vérifier et noter le fonctionnement de votre système (XXXXXX).

Pour la partie systèmes distribués, en plus du rapport, vous fournirez le code source de vos programmes.

Toute tricherie impliquera un 0. Les correcteurs se réservent le droit de donner une note différenciée à chaque membre d'un binôme.