

Rapport de projet DDS

version AE-23/02/2021

*Mettez ici vos noms et les parties sur lesquelles vous avez travaillées :
conception, robot, vidéo, mission, intégration, rédaction du compte-rendu
Exemple : P.-E. Hladik (conception, robot, rédaction du compte-rendu)*

— Les attendus du projet —

Pour ce travail, il est attendu que :

- vous réalisez une conception de l'application en entier avec les parties communication, robot et vidéo (voir partie 1 du rapport pour plus de détails),
- vous implémentiez uniquement les parties communication et robot (fonctionnalités 1 à 13),
- vous fassiez une vidéo qui démontre le fonctionnement de votre application (voir partie 3).

Il est possible d'implémenter la partie vidéo, cependant ce travail n'est pas attendu et ne sera donc pas pris en considération dans l'évaluation. C'est cependant possible de le faire...

— Ce qu'il faut faire pour le rapport —

Remplacez tous les textes en bleu et supprimer les textes en rouge

Le document à rendre par groupe doit être une archive zip ou tar. Cette archive sera à déposer sur moodle avant le 26 mars 2021 23h59 : <https://moodle.insa-toulouse.fr/mod/assign/view.php?id=7360>

L'archive doit contenir quatre fichiers :

- **votre rapport en format pdf,**
- **vos fichier `tasks.h` et `tasks.cpp`,**
- **un fichier vidéo avec vos tests.**

Vous pouvez utiliser word ou un autre logiciel d'édition pour rédiger ce rapport, par contre vous devez impérativement respecter la structure ci-après et le rendre au format pdf.

Remarque : si vous n'arrivez pas à déposer la vidéo sur moodle, faites un dépôt sur un

drive (ou autre) et ajoutez le lien dans le rapport.

Critères d'évaluation :

- Qualité rédactionnelle,
- Exhaustivité et justesse des règles de codage,
- Qualité de la conception (clarté, respect de la syntaxe, exhaustivité, justesse),
- Qualité des explications pour la validation,
- Respect des règles dans la production du code.

Compétences évaluées :

- rédaction et communication sur un dossier de conception,
- concevoir une application concurrente temps réel,
- passer d'un modèle de conception à une implémentation,
- écriture de code et utilisation de primitives au niveau système,
- valider un code.

1 Conception

La conception doit couvrir l'ensemble des fonctionnalités du cahier des charges.

Mettez dans cette partie tous les éléments de votre conception en particulier vos diagrammes AADL (vue globale du système) et les diagrammes d'activité (détails des threads). Cette partie doit être auto-suffisante pour comprendre votre application.

Pour faciliter la lecture des schémas, vous allez présenter votre conception en trois parties, l'une focalisée sur la communication entre le moniteur et le superviseur, la seconde consacrée au traitement vidéo et la troisième au contrôle du robot.

Si vous le souhaitez, au lieu de dessiner vos diagrammes sous un éditeur, vous pouvez joindre un scan de vos schémas — ils doivent être lisibles et propres.

1.1 Diagramme fonctionnel général

Mettez ici un diagramme fonctionnel qui présente les principaux blocs de votre conception. Pour cela, inspirez vous du diagramme ci-dessous (fig. 1.1) en indiquant pour chaque groupe de threads les données et ports partagés. La figure 1.1 a été réalisée à partir du document de conception. **Vous devez absolument conserver le découpage en trois groupes de threads** (`th_group_gestion_moniteur`, `th_group_vision`, `th_group_gestion_robot`).

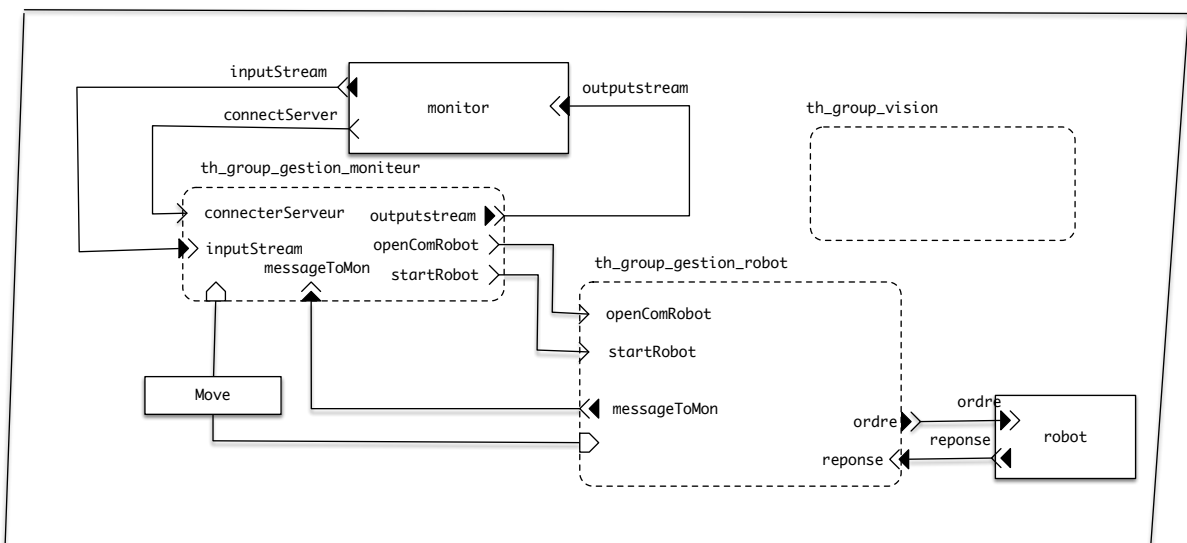


Fig. 1: Diagramme fonctionnel du système

1.2 Groupe de threads gestion du moniteur

Placez ici :

- le diagramme fonctionnel en AADL décrivant le groupe de threads de gestion du moniteur (voir exemple de la figure 1.2.1 réalisée à partir du dossier de conception),
- les diagrammes d'activité de chaque thread de ce groupe.

Décrivez tous les éléments (paramètres, variables, etc.) qui vous semblent pertinents pour comprendre les diagrammes.

1.2.1 Diagramme fonctionnel du groupe gestion du moniteur

Exemple de diagramme fonctionnel pour le groupe de thread de gestion du moniteur. Mettez à jour ce diagramme avec votre conception.

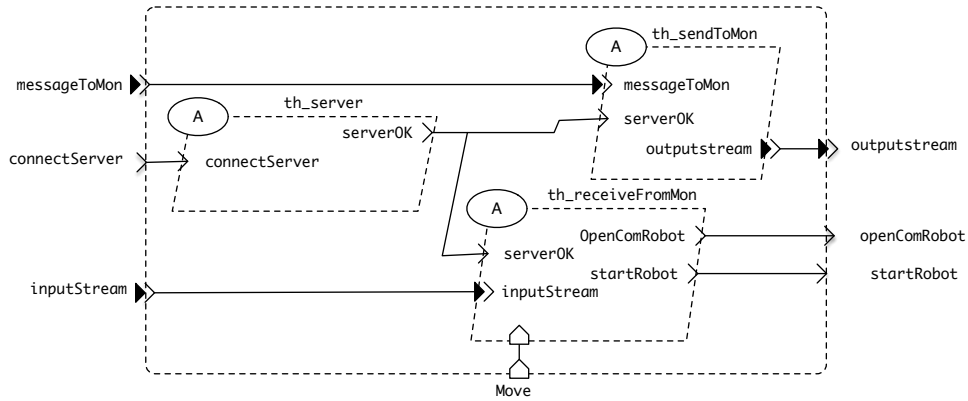


Fig. 2: Diagramme fonctionnel du groupe de threads gestion du moniteur

1.2.2 Diagrammes d'activité du groupe gestion du moniteur

Décrivez le comportement de chacun de vos threads avec des diagrammes d'activité. Apportez les explications qui vous semblent nécessaires pour comprendre votre conception. A titre d'exemple les diagrammes fonctionnels tirés du document de conception sont remis.

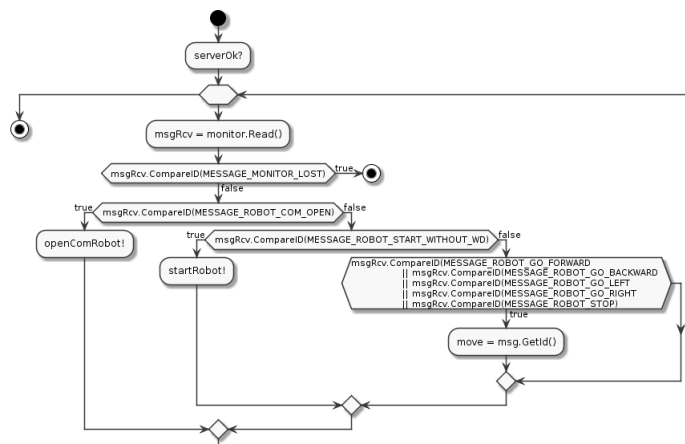
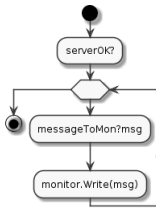
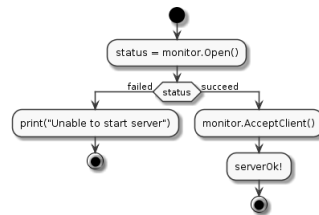


Fig. 3: Diagramme d'activité du thread `th_receiveFromMon`

Fig. 4: Diagramme d'activité du thread `th_sendToMon`Fig. 5: Diagramme d'activité du thread `th_server`

1.3 Groupe de threads gestion du robot

1.3.1 Diagramme fonctionnel du groupe gestion robot

Ajoutez le diagramme fonctionnel du groupe de threads de gestion du robot.

1.3.2 Diagrammes d'activité du groupe robot

Décrivez le comportement de chacun de vos threads avec des diagrammes d'activité. Apportez les explications qui vous semblent nécessaires pour comprendre votre conception.

1.4 Groupe de threads vision

1.4.1 Diagramme fonctionnel du groupe vision

Ajoutez le diagramme fonctionnel du groupe de threads de vision.

1.4.2 Diagrammes d'activité du groupe vision

Décrivez le comportement de chacun de vos threads avec des diagrammes d'activité. Apportez les explications qui vous semblent nécessaires pour comprendre votre conception.

2 Transformation AADL vers Xenomai

Cette section est consacrée à la méthode pour passer d'un modèle AADL à un code sous Xenomai. Pour chacun des éléments AADL, vous expliquerez **comment vous l'avez traduit en code** et quels **services de Xenomai** vous avez utilisés **en expliquant ce qu'ils font**. Chaque élément devra être illustré avec des **extraits de code de votre projet**.

2.1 Thread

2.1.1 Instanciation et démarrage

Expliquer comment vous implémentez sous Xenomai l'instanciation et le démarrage d'un thread AADL.

Exemple de réponse : Chaque thread a été implémenté par un `RT_TASK` déclarés dans le fichier `tasks.h`. La création de la tâche se fait à l'aide du service `rt_task_create` et son démarrage à l'aide de `rt_task_start`. Toutes les tâches sont créés dans la méthode `init` de `tasks.cpp` et démarrées dans la méthode `run`.

Par exemple, pour la tâche `th_server`, sa déclaration est faite ligne 73 dans le fichier `tasks.h`

```
RT_TASK th_server;
```

sa création ligne 102 de `tasks.cpp` lors de l'appel de

```
rt_task_create(&th_server, "th_server", 0, PRIORITY_TSERVER, 0)
```

et son démarrage ligne 146 avec

```
rt_task_start(&th_server, (void*)(void*)) & Tasks::ServerTask, this)
```

2.1.2 Code à exécuter

Comment se fait le lien sous Xenomai entre le thread et le traitement à exécuter.

2.1.3 Niveau de priorités

Expliquer comment vous fixez sous Xenomai le niveau de priorité d'un thread AADL.

2.1.4 Activation périodique

Expliquer comment vous rendez périodique l'activation d'un thread AADL sous Xenomai.

2.2 Donnée partagée

2.2.1 Instanciation

Quelle structure instancie une donnée partagée ?

2.2.2 Accès en lecture et écriture

Comment garantes-vous sous Xenomai l'accès à une donnée partagée ?

2.3 Port d'événement

2.3.1 Instanciation

Comment avez-vous instancié un port d'événement ?

2.3.2 Envoi d'un événement

Quels services ont été employés pour signaler un événement ?

2.3.3 Réception d'un événement

Comment se fait l'attente d'un événement ?

2.4 Ports d'événement-données

2.4.1 Instanciation

Donnez la solution retenue pour implémenter un port d'événement-données avec Xenomai.

2.4.2 Envoi d'une donnée

Quels services avez-vous employés pour envoyer des données ?

2.4.3 Réception d'une donnée

Quels services avez-vous employés pour recevoir des données ?

3 Analyse et validation de la conception

Cette partie sera à réaliser par la production d'une vidéo dans laquelle vous montrerez que chacune des fonctionnalités de 1 à 13 (communication et robot) sont opérationnelles. Vous n'êtes pas obligé de reprendre les fonctionnalités déjà faites (1,2,3, 4, 7, 10 et 12) sauf si vous les avez modifiées.

Faites une courte séquence vidéo pour chaque fonctionnalité et montrer clairement qu'elle respecte les besoins. Pour réaliser la vidéo, enregistrer votre écran en commentant. Une vidéo d'exemple (rapidement produite et avec le simulateur) pour les fonctionnalités (1,2,3, 4, 7, 10 et 12) est postée sur <https://drive.google.com/open?id=1Dk5C9FLq-bFZCgSRN2y9dTck0oxM7fp4>