

MI11 - TP modélisation AADL et ordonnancement

15 juin 2017

1 Mise en place

Ce TP exploite l'environnement Eclipse (en particulier le plug-in OSATE destiné à la modélisation et l'analyse AADL) et le logiciel d'analyse et de simulation d'ordonnabilité Cheddar.

1. Naviguez jusqu'au dossier `C : \MI11\OSATE2`
2. Lancez `osate.exe`
3. Choisissez un dossier sur votre disque Z pour stocker l'espace de travail (par exemple `Z : \aadl`)
4. Créez un nouveau projet de type "AADL/AADL Project" à partir du menu "File/New/Project..."
5. Créez un lanceur pour Cheddar :
 - Dans le menu "Run", choisissez "External Tools" puis "External Tools Configurations..."
 - Créez une nouvelle configuration dans la catégorie "Program", que vous appellerez "Cheddar"
 - Dans "Location", entrez `C : \MI11\Cheddar-3.0\cheddar.exe`
 - Dans "Working directory", entrez `C : \MI11\Cheddar-3.0\`
 - Dans "Arguments", insérez le texte suivant :
 - I "C : \MI11\Cheddar-3.0\project_examples\aadl"
 - a "\${selected_resource_loc}"
 - Cliquez sur "Run"
 - Vérifiez que Cheddar est bien lancé puis fermez-le.
6. Importez dans le projet les fichiers source AADL fournis sur le site Moodle de l'UV.

Pour faire une analyse, sélectionnez le fichier AADL correspondant et lancez "Run/External Tools/Cheddar". Cheddar importe alors le modèle AADL et est prêt pour l'analyse et la simulation.

2 Cheddar et AADL

2.1 Modèle des tâches

La figure suivante rappelle les paramètres temporels essentiels d'une tâche.

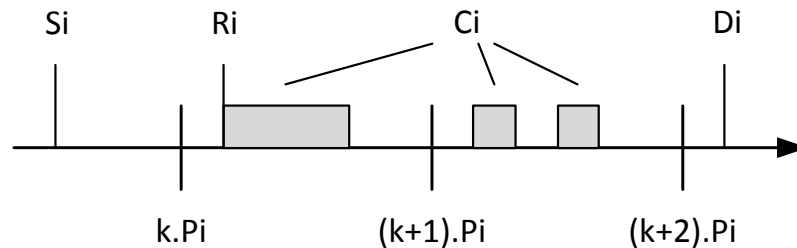


FIGURE 1 – Paramètres d'une tâche

Les tâches peuvent être :

- Périodiques. La tâche est activée répétitivement à chaque nouvelle période. Elle est définie par sa date d'arrivée dans le système S_i , sa capacité C_i , sa période d'activation P_i et son délai critique relatif au début de la période D_i .
- Apériodiques. La tâche est activée une seule fois. Elle est définie par sa date d'arrivée dans le système S_i , sa capacité C_i , sa date d'exécution au plus tôt R_i et son délai critique D_i .
- Sporadiques. La tâche est définie de manière similaire à une tâche périodique, mais P_i constitue le délai minimum entre deux activations.

2.2 Propriétés AADL associées

Ces paramètres sont exprimés dans un modèle AADL sous forme de propriétés associées aux *threads* qui représentent les tâches.

- `Period` – période P_i
- `Compute_Execution_Time` – capacité C_i . La valeur de ce paramètre est un intervalle temporel (par exemple 1 ms .. 5 ms). La valeur maximale (Worst Case Execution Time) est retenue pour l'analyse.
- `Deadline` – Délai critique D_i
- `Cheddar_Properties::Dispatch_Absolute_Time` – Date d'arrivée S_i
- `Dispatch_Protocol` – Protocole d'activation de la tâche :
 - `Periodic` – Tâche périodique
 - `Background` – Tâche apériodique
 - `Sporadic` – Tâche sporadique
- `Cheddar_Properties::Fixed_Priority` – Priorité fixe entre 1 et 255 (255 est la plus haute priorité) pour les politiques d'ordonnancement telles que *Rate Monotonic*.

La politique d'ordonnancement est une propriété des processeurs :

- `Cheddar_Properties::Scheduling_Protocol` – Politique d'ordonnancement appliquée sur le processeur associé :
 - `RATE_MONOTONIC_PROTOCOL`
 - `EARLIEST_DEADLINE_FIRST_PROTOCOL`
 - `LEAST_LAXITY_FIRST_PROTOCOL`
 - `POSIX_1003_HIGHEST_PRIORITY_FIRST_PROTOCOL`
- `Cheddar_Properties::Preemptive_Scheduler` – Indique si l'ordonnanceur est pré-emptif ou non (True ou False).

L'intégralité des propriétés liées à Cheddar sont documentées sur le site <http://beru.univ-brest.fr/~singhoff/cheddar/publications/singhoff-03-2007.html>.

3 Travail à réaliser

Sauf précision contraire, l'unité de temps est la milliseconde (ms). Dans votre rapport, vous donnerez vos codes ainsi que les copies des écrans Cheddar qui vous permettent d'arriver à vos conclusions (ordonnancements, résultats, ...)

Exercice 1 : Ordonnancement Rate Monotonic

Soient trois tâches périodiques T1, T2 et T3 définies par les paramètres suivants :

Tâche	S_i	P_i	C_i	D_i
T1	0	29	7	29
T2	0	5	1	5
T3	0	10	2	10

- Q1. Calculez le taux d'utilisation du processeur. Le jeu de tâche est-il ordonnançable ?
- Q2. Donnez les priorités respectives des tâches T1, T2 et T3.
- Q3. Modifiez le fichier `tp_rm.aadl` avec les paramètres des tâches et de l'ordonnanceur.
- Q4. Observez les deux ordonnancements générés par Cheddar, d'abord pour la version préemptive puis non préemptive. Que constatez-vous ?
- Q5. On suppose maintenant que $P_1 = 30$ et $C_1 = 6$ et que $C_2 = 3$, pour former un jeu de tâches harmoniques (les périodes sont multiples entre elles). Refaites le test d'ordonnançabilité puis observez l'ordonnancement par simulation sur la période d'étude. Que constatez-vous ?
- Q6. Confirmez le résultat précédent au moyen des temps de réponse des tâches donnés par Cheddar.

Exercice 2 : Ordonnancement EDF

Soient trois tâches périodiques T1, T2 et T3 définies par les paramètres suivants :

Tâche	S_i	P_i	C_i	D_i
T1	0	12	5	12
T2	0	6	2	6
T3	0	24	5	24

- Q1. Calculez le taux d'utilisation du processeur. Le jeu de tâche est-il ordonnançable ?
- Q2. Dans le projet, créez une copie du fichier `tp_rm.aadl`, que vous appellerez `test_edf.aadl`. Modifiez ce fichier pour paramétrer correctement les threads et le processeur.
- Q3. Calculez le nombre d'unités de temps libre sur la période d'étude.
- Q4. Confirmez le calcul précédent au moyen de Cheddar en simulant le système sur la période d'étude, d'abord en version préemptive, puis en version non préemptive.
- Q5. Soient deux tâches aperiodiques TA1 et TA2, de date d'arrivée S_i respectives 7 et 12 et de capacités C_i respectives 1 et 3. Modifiez le modèle AADL pour inclure ces deux tâches. Que peut-on dire de l'ordonnançabilité de ce nouveau jeu de tâches en mode préemptif ?

Exercice 3 : Moniteur médical multiparamètres

Dans cet exercice, on étudie un système de surveillance de patient multiparamètres. Ce système réalise l'acquisition de trois paramètres vitaux (la saturation sanguine SO₂, la pression artérielle PA et l'activité cardiaque ECG) d'un patient. Il les affiche en permanence sur un écran LCD intégré et peut déclencher une alarme si l'un d'entre eux sort des bornes normales.

La partie logicielle représentée par le processus `health_monitoring` est constituée de cinq traitements :

1. La tâche *getPA* lit toutes les 10 ms la pression artérielle
2. La tâche *getSO₂* lit toutes les 10 ms la saturation sanguine
3. La tâche *getECG* lit toutes les 40 ms l'activité cardiaque du patient.
4. La tâche *dpy* produit toutes les 6 ms un affichage sur l'écran LCD
5. La tâche *check* vérifie les paramètres toutes les 12 ms.

Les durées d'exécutions des tâches *getSO₂* et *check* sont inférieures à 2 ms. Les tâches *getPA* et *dpy* ont une durée inférieure à 1 ms. Enfin, la tâche *getECG* a besoin de 4 ms pour collationner toutes les données relatives à l'activité cardiaque.

Les tâches s'exécutent dans un environnement POSIX. L'ordonnanceur est donc préemptif à priorité fixe. Deux tâches ne peuvent pas avoir le même niveau de priorité.

- Q1. Sur quels critères peut-on affecter la priorité des tâches en règle générale ? Déterminer les différents paramètres des tâches nécessaires à leur ordonnancement, en justifiant les priorités que vous avez décidé d'affecter.
- Q2. Implémentez ces paramètres au moyen des propriétés AADL appropriées dans le modèle `health_monitor.aadl`. Vous veillerez à configurer correctement l'ordonnanceur pour appliquer la politique `POSIX_1003_HIGHEST_PRIORITY_FIRST_PROTOCOL`.
- Q3. Vérifiez l'ordonnabilité au moyen de Cheddar sur la période d'étude.

Afin d'augmenter les performances du système, on veut maintenant répartir les tâches sur deux processeurs (les processeurs `cpu_a` et `cpu_b`). Elles sont maintenant définies par les paramètres suivants :

Tâches	Processeur	Capacité	Période
check	b	4	6
getECG	b	2	20
dpy	a	2	3
getPA	a	2	5
getSO ₂	a	2	5

Les tâches restent ordonnancées selon la même politique que précédemment. De plus, les priorités sont déterminées selon l'algorithme Rate Monotonic.

- Q4. Toutes les tâches respectent-elles encore leurs contraintes temporelles avec la répartition proposée dans la table ?
- Q5. L'affectation des tâches aux processeurs a et b a été réalisée aléatoirement. Les tâches peuvent en effet être allouées indifféremment à l'un des deux processeurs. Sachant que le processeur b est deux fois plus rapide que le processeur a, proposez une nouvelle affectation qui permet à toutes les tâches de respecter leurs contraintes temporelles en étudiant leurs périodes respectives et en vous appuyant sur les résultats de l'exercice 1. N'oubliez pas de mettre à jour les capacités des tâches en fonction des performances du processeur sur lequel elles sont allouées en vous basant sur leurs capacités pour l'allocation proposée dans la table. Justifiez votre choix.
- Q6. Modifiez le modèle AADL du système afin de refléter ces changements (ajout de processeurs, de process, nouvelle décomposition du logiciel et affectation). Donnez avec Cheddar un chronogramme de l'ordonnancement sur la période d'étude.