M1 ALMA
Université de Nantes
2010-2011

Projet de Travaux pratiques : Systèmes Distribués Mini -projet1

MARGUERITE Alain
RINCE Romain

Université de Nantes
2 rue de la Houssinière, BP92208, F-44322 Nantes cedex 03, FRANCE

Encadrant : QUEUDET Audrey

Table des matières

1 Cahier des charges

Introduction : L'objectif est de définir un outil de simulation d'ordonancement de tâche en temps réel. Parmies ses fonctionalités, l'outil devra pour tester les contraintes temporelles d'un ensemble de tâches générées au préalable. La génération de ces tâches entre par dans la conception de l'outil. Cet outil permettra d'exporter le résultat dans un fichier d'extension *.ktr* pour être exploité directement par l'outil graphique Kiwi.

1.1 Données en entrées

L'outil doit pouvoir permettre à l'utilisateur de rentrer des tâches périodiques et ou apériodiques lui même (en précisant chacune des attributs) ou de demander une génération aléatoire pour les deux catégories.

1.2 Fonctionnement

- Une analyse d'ordonnançabilité. L'outil affichera à l'utilisateur les résultats des différents test s'afficheront avec les conclusions qui en découlent.
- Un environnement de simulation. L'outil lors du calcul de de l'ordonancement devra afficher les différents événements. Un bilan de ces action sera résumé dans un fichier au terme de l'execution (facultatif).
  L'outil doit pouvoir proposer plusieurs politiques d'ordonnancement. Á savoir :
  - Pour les tâches périodiques :
    - Rate Monotonic
    - EDF
  - Pour les tâches apériodique :
    - BG
    - TBS
- Un fichier d'extension *.ktr* sera généré au terme de l'execution, et contiendra le déroulement de l'ordonancement jusqu'a son terme.

- L'outil doit communiquer au terme de l'execution différents résusltats de performance qu'il aura lui même calculés. Les informations à fournir sont les suivantes :
  - Le nombre de violations d'échéances.
  - le nombre de commutations de contexte et de préemptions.
  -

2 Analyse et solutions du problème

2.1 Introduction et choix du langage

Dans le cadre du module Systèmes Distribués, l'opportunité de spécifier et concevoir un générateur de tâches temps réel nous est proposé. Le langage d'implémentation étant libre, notre binome a opté pour Java. Ce choix est basé principalement par le fait qu'il s'agissait du langage le plus maitrisé. Cela nous a permis de concentrer nos efforts sur la mise en place des algorithmes et non sur des probèmes de langage. L'objectif est de définir un outil de simulation d'ordonancement de tâche en temps réel.

2.2 Modelisation du problème

Le cahier des charges demandait une gestion de taches périodique et apériodiques. blah blah blah d'ordonancement

2.3 Génération de tâches dans un fichier

La première partie du projet avait pour objectif d'obtenir un nombre n de taches périodiques et apériodiques pour de futurs traitements décrits dans la partie 2 : rajouter lien dym. Á nouveau le choix du format d'un tel fichier nous était laissé. Nous avons choisis de générer un fichier xml (à nouveau pour des raisons de simplicité) à la syntaxe suivante :

- Des balises `<genTache.AbstractTache-array>` encadrent la totalité du fichier.
- Une tâche périodique sera définie dans une balise `<genTache.TachePeriodique>`
- Une tâche apériodique sera définie dans une balise `<genTache.TacheAPeriodique>`
- Dans une tache tous ses attributs seront définis de la manière suivante `<nom_attribut>valeur_attribut</nom_attribut>`

Voici un exemple d'un respectant le format décrit ci-dessus :

```
<genTache.AbstractTache-array>
  <genTache.TachePeriodique>
    <Pi>377</Pi>
    <ri>0</ri>
    <id>1</id>
    <Ci>1</Ci>
    <Di>1</Di>
  </genTache.TachePeriodique>
  <genTache.TachePeriodique>
    <Pi>162</Pi>
    <ri>0</ri>
    <id>2</id>
    <Ci>6</Ci>
    <Di>30</Di>
  </genTache.TachePeriodique>
  <genTache.TacheAperiodique>
    <ri>859</ri>
    <id>3</id>
    <Ci>26</Ci>
    <Di>71</Di>
  </genTache.TacheAperiodique>
</genTache.AbstractTache-array>
```

On remarque que les taches périodiques sont identifiées par :

- Pi : période d'activation.
- ri : date de réveil.
- id : L'id de la tâche.
- Ci : durée d'exécution maximale.
- Di : délai critique

Alors que les tâches apériodiques ont seulement :

- ri : date de réveil
- id : L'id de la tâche.
- Ci : durée d'exécution maximale.
- Di : délai critique.

2.4 Algorithmes et structures mises en oeuvre

2.4.1 Première Partie

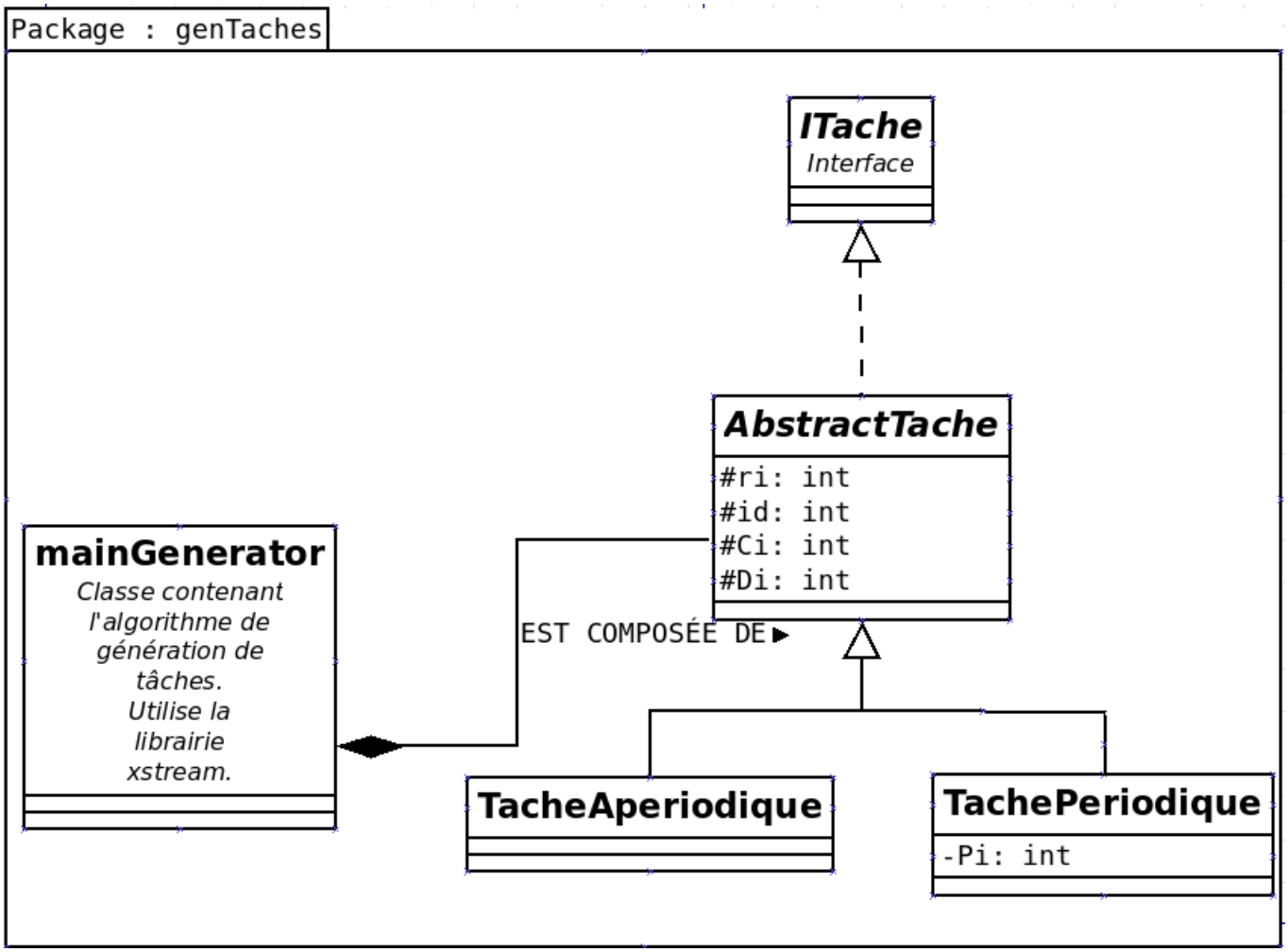

FIGURE 2.1 – Package générateur de tâches temps réel

Calcul des tâches ap Le calcul des tâches ap est effectué selon la formule suivante : $U_a = \frac{\sum_{i=1}^{m} C_i}{ppcm(P_i)}$ .... L'utilisateur entre la variable Uap et le nombre de tache ap qu'il désire (m) les variables restantes charge des ap sur une hyperpériode

2.4.2 Deuxième partie

La deuxième partie du projet, consistait à partir du fichier de tâches (cf Première Partie), de creer un simulateur proposant deux types de fonctionnalités :

- une analyse d'ordonnançabilité,
- un environnement de simulation.

Le premier sujet de reflexion était donc dans quelles structures de donnée stocker les tâches et par quels moyens. Le choix des ArrayList s'est fait naturellement grâce en premier plan à sa facilité de manipulation. Par ailleurs la fonctionnalité de la bibliothèque xtream permettant d'extraire le contenu d'un fichier xml dans une ArrayList en quelques lignes nous a d'emblée convaincu. Une fois cette première difficulté franchie une seconde de taille est apparue. Nous avons compris qu'il serait délicat de modifier directement les tâches dans les ArrayList. Notre idée initiale était de modifier son Ci d'une tâche après sont traitement dans l'algorithme. Ce qui dans le cas des tâches périodique par exemple, serait problématique. En effet par définition les taches périodiques sont suseptible à être traitésà plusieurs reprise. Il faut donc garder leurs propiétés initiales intactes. Nous avons donc choisi d'organiser notre programmation en plusieurs classes pour bien séparer les opérations de manipulation des données d'une part, les opérations mettant en jeux les algorithmes d'ordonancement et enfin celles écrivant dans le fichier de sortie au format .ktr

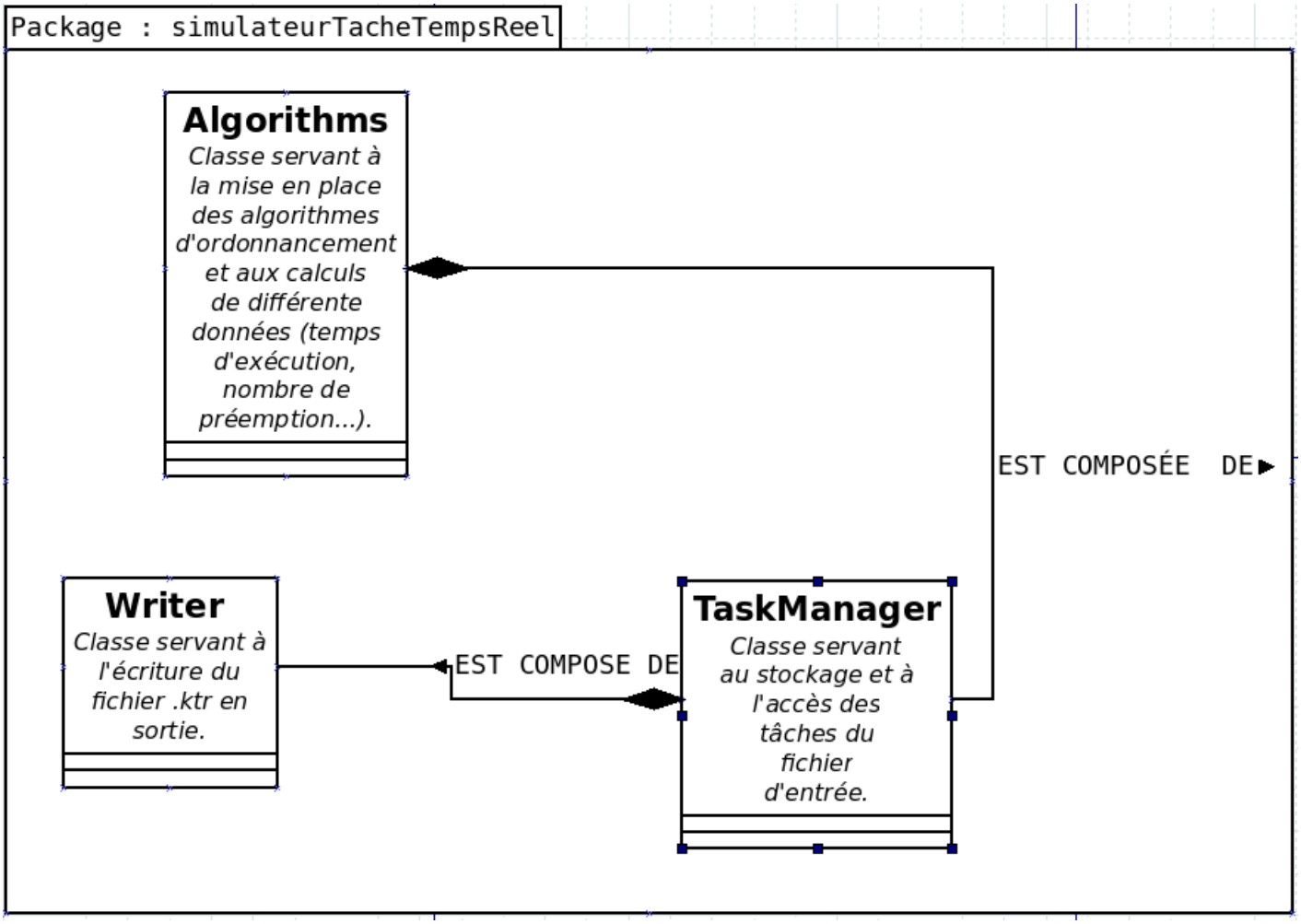

FIGURE 2.2 – Package : Simulateur de systèmes temps réel

2.5 Interface proposée

Toutes l'interface de l'outil est en ligne de commande. L'absence d'IHM graphique est due au manque de temps et au peu d'intêret que cela représentait pour le problème étudié. De même l'outil ne prend en compte aucune option. Même si cet aspect aurait apporté un certain confort à l'utilisateur, nous avons opté pour la solution de lui demandé au fur et à mesur de l'execution du programme les diverses options et paramètres requis. Nous avons regretté ce choix lors de la phase des test (la répétitions sans cesse des divers option est fastidieuse). Cependant nous avons gagné un temps de conceptio non négligeable.

3 Manuel utilisateur

3.1 Introduction et objectifs

3.1.1 Avis au lecteur

Ce manuel est destiné à un public désirant utiliser le logiciel GT. C'est à dire depuis son installation jusqu'à la génération du fichier au d'un fichier au format .ktr.Le manuel n'a pas pour objectif d'enseigner l'utilisation de cet outil. Les auteurs recommandes l'ouvrage suivant pour un tel apprentissage : [kiw].

3.1.2 Présentation du logiciel GT

GT permet la génération automatique ou manuel de de systèmes temps réel. L'outil est dédié est uniquement utilisable en mode console.

3.2 Votre première simulation

3.2.1 Introduction

Ce chapitre va vous permettre de réaliser une vote premierès simulation pour la validation de systèmes temps réel en moins de 2 minutes ☺ ! Voici le résultat que vous obtiendrez au terme :

FIGURE 3.1 – Fichier .ktr

3.2.2 1ère Étape : Génération des tâches avec GT

Dans le dossier GT, entrez la commande suivante `./GT`. On vous propose d'entrez un nom de fichier. Entrez un nouveau nom si vous voulez créer vos propres tâches, ou entrez cours_RMBG si vous voulez utiliser un exemple déjà éxistant. Il vous sera demandé par la suite quel type d'ordonnancement vous désirez utiliser. Tapez R pour Rate Monotonic-Background.

3.2.3 2ème Étape : Utilisation de Kiwi

Placez vous dans le dossier ou est installé Kiwi. Tapez la commande suivante : `./kiwi &`. Une fois dans l'interface de Kiwi, cliquez sur Open. Allez ouvrir votre fichier .ktr se trouvant dans le dossier ou est installé GT. Cliquez sur l'îcone Play. Voila ce que vous obtenez :

FIGURE 3.2 – Exemple d'utilisation de GT

Et voilà avez crée votre première simulation pour la d'un systèmes temps réel ! Si vous avez rencontrez des difficultés au cours de ce chapitre, vous pouvez vous référer aux différentes parties de ne manuel qui détaille chaque étapes en détail.

3.3 Installation et configuration

3.3.1 Configuration nécessaire

⚠ : GT est un outil open source dédié uniquement aux systèmes d'exploitation linux. Il n'existe pas encore de version pour Windows et MAC OS. L'installation requière des connaissances dans la manipulation de commandes shell. L'ouvrage suivant est une référence dans ce domaine : [EAS+05]

3.3.2 Installation

Rendez vous sur http://www.GT.org. La rubrique «Download» vous proposera une archive de type tar.gz pour différentes distributions (solaris, Linux 32 Bit, Linux 64 Bit, ...). Le téléchargement terminé, décompressez l'archive dans le dossier où vous désirez installer GT. Placez vous dans ce dossier et tapez la commande `java -jar GT`. GT est maintenant installé sur votre ordinateur ☺!!

3.4 Utilisation par un terminal

GT peut être utilisé uniquement en ligne de commande. Cette partie demandes des connaissances pré requises sur les commandes unix. En effet seul les fonctionalités de l'outil seront explicitées. La mecanisme des options est similaire à toute autres commandes unix. Pour plus de d'information sur les système Unix , nous recommandons l'ouvrage suivant : [DMK+06].

3.4.1 Utilisation basique

La simple commande suivante executera l'outil GC `./GT` vous donne la possibilité d'entez un nom de fichier. Vous avez alors deux possibilité :

- `circulaire` Entrez un nom de fichier .xml déjà éxistant dans le dossier courant. Le fichier .ktr sera crée à partir de ce fichier
- `histogramme` Entrez un nouveau nom et suivez les instruction pour la génération de tâches

3.4.2 Génération manuelle des tâches

Il vous sera demandé successivement si vous souhaitez une génération automatique ou des nom des tâches périodiques. Dans le cas d'une génération automatique il est necessaire de renseignerle pourcentage d'utilisation maximale du processeur. Pour les tâches apériodiques il est aussi necessaire de fournir leur taux d'occupation du processeur (U_a). Dans le cas d'une génération entièrement manuelle, il est demandé de fournir charque caractéristiques de la tâche. Pour plus de détails sur chaque attribut à fournir nous vous invitons à vous référer au document suivant : [Que]

3.5 Copyright

Ce programme est distribué dans l'espoir qu'il sera utile, mais SANS AUCUNE GARANTIE ; sans même la garantie implicite de QUALITÉ MARCHANDE ou D'ADÉQUATION À UNE UTILISATION PARTICULIÈRE. Pour plus de détails, reportez-vous à la GNU General Public License.

Vous devez avoir reçu une copie de la GNU General Public License avec ce programme. Si ce n'est pas le cas, consultez [GNU]

4 Jeux d'essais

Jeux d'essaisJeux d'essaisJeux d'essaisJeux d'essaisJeux d'essaisJeux d'essaisJeux d'essais-Jeux d'essaisJeux d'essaisJeux d'essaisJeux d'essaisJeux d'essaisJeux d'essais

5 Bilan et conclusion

parler qu'une autre structure que les ArrayList aurait été mieux. bilan

Table des figures

Bibliographie

[DMK+06] Dalheimer, Matthias, Kalle, Welsh, and Matt. *Le système Linux*, volume 1. O'Reilly, 5ème édition edition, 2006.

[EAS+05] E.Siever, A.Weber, S.Figgins, R.Love, and A.Robbins. *Linux in a Nutshell*. O'Reilly Media, 5ème édition edition, Juillet 2005.

[GNU] Gnu, operating system. http://www.gnu.org/licenses/.

[kiw] Universidad politécnica de valencia. http://www.gti-ia.upv.es/sma/tools/kiwi/.

[Que] Audrey Queudet. Linux pour le temps-réel. http://www.google.com/url?sa=t&rct=j&q=os%20temps%20r%C3%A9el%20ordonnacement%20audrey&source=web&cd=1&ved=0CCYQFjAA&url=http%3A%2F%2Fpagesperso.lina.univ-nantes.fr%2Finfo%2Fperso%2Fpermanents%2Fqueudet%2Fdocs%2FModule_E4_Linux_Temps_Reel.pdf&ei=0ltJT7zHC82q8APEpYWZDg&usg=AFQjCNHoXlYj01u8FNg7cm0UIYWNG7K23A&cad=rja.