Martin Thomas

Khaled Arsalane

ITR : Visualiseur d'ordonnancement

Heptane :

Les exécutables sur lesquels nous avons utilisés Heptane sont simplistes. Rencontrant quelques difficultés à obtenir des résultats satisfaisant avec le robot, nous avons du nous en contenter, le temps pour produire le visualiseur se réduisant. Le fichier xml que nous avons obtenu a malheureusement été le seul et unique jeu de test à disposition, cependant, notre projet est censé pouvoir gérer un nombre de tâche arbitraire, du moment que toutes les informations sont passées en paramètres (cf Ordonnanceur :).

Ordonnanceur :

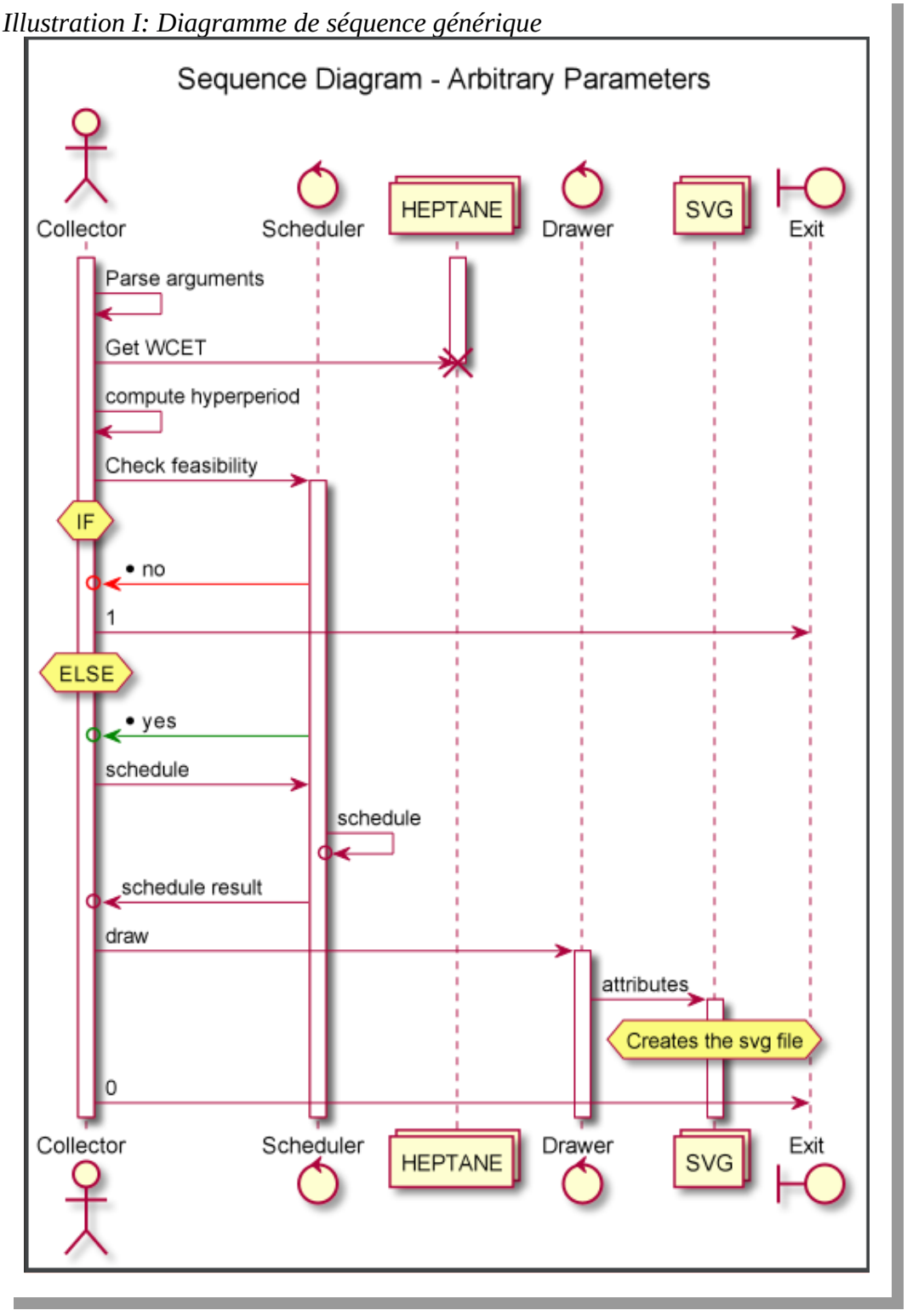
Afin de pouvoir ordonnancer les tâches selon les algorithmes Rate Monotonic et Earliest Deadline First, le programme nécessite certaines informations dont :

- Le nom de l'algorithme
- Le WCET des tâches, récupérés depuis le fichier Heptane
- Le nom des tâches auxquelles on s'intéresse
- Leur période
- Leur Deadlines (dans le cas de l'algo EDF)

Ces données sont pour la plupart récupérés depuis les paramètres de la commande(cf Utiliser le visualiseur :). Une fois ces informations

regroupées, le visualiseur les soumettra à l'ordonnanceur, qui produira les listes ordonnées de l'exécution des tâches intéressantes.

Illustration I: Diagramme de séquence générique



Génération du dessin :

L'ordonnanceur ayant produit les listes ordonnées de l'exécution des tâches, il nous a suffi de représenter les tâches sous forme de rectangles dont la dimension représente la durée d'exécution en ms.

Un souci d'échelle nous a forcé à réduire la taille des élements, d'où la présence des constantes ECHELLE dans drawer.py et SCALE dans collector.py Les résultats obtenus correspondent visuellement à

ce que nous voulions obtenir. Exemples non exhaustifs ci-dessous.

Illustration III: Rate Monotonic non préemptif

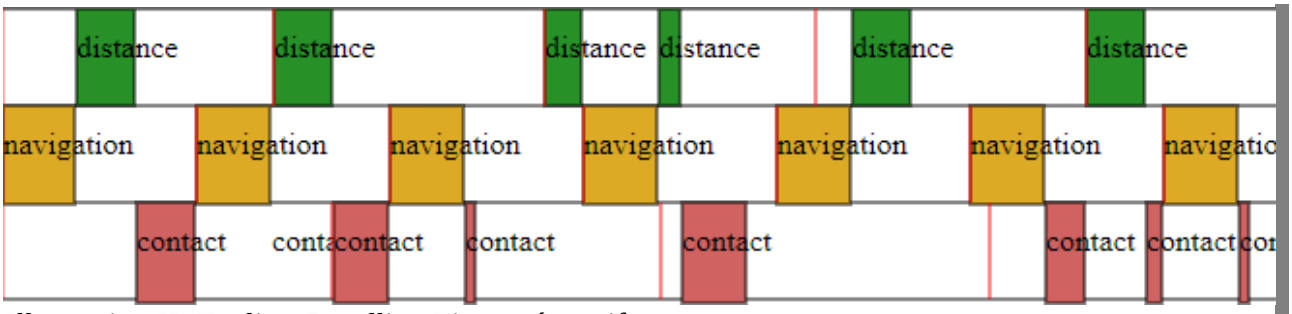


Illustration II: Earliest Deadline First préemptif

Utiliser le visualiseur :

Pour lancer le visualiseur, nous vous recommandons la version de python 3.9, version avec laquelle nous avons développé ce projet. Il vous faudra également la librairie svgwrite, à installer avec pip si ce n'est pas déjà fait.
L'argument help de la commande est fonctionnel pour avoir les détails des arguments.
Un README,md est également disponible dans les sources et sur le Github[1] de ce projet , vous y trouverez des exemples de lancement. Les tâches de notre exécutables sont contact, distance et navigation.
Architecture du code ci-contre.

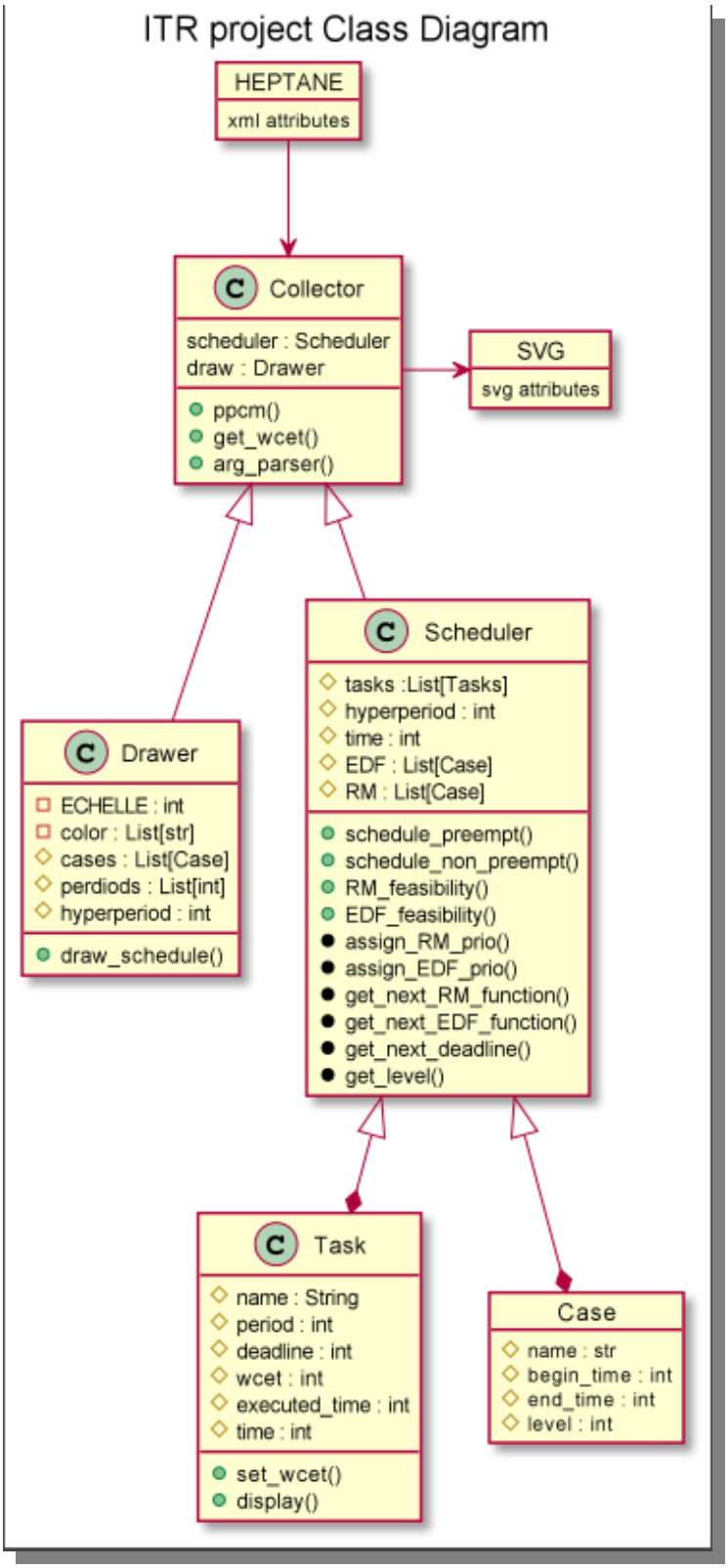


Illustration IV: Diagramme de classe du projet

1 https://github.com/bibi14010/itr-scheduler-2020