

PCD ASS02 (1)

Federico Bravetti

Tommaso Patriti

Alex Testa

federico.bravetti2@studio.unibo.it tommaso.patriti@studio.unibo.it alex.testa@studio.unibo.it

I. ANALISI DEL PROBLEMA

A partire dalla soluzione proposta nell'Assignment1 si vuole modificare il programma andando a delegare la gestione dei task al framework *Executor*.

II. STRATEGIA RISOLUTIVA

Il framework *Executor* offre un supporto alla programmazione task-oriented, per cui avendo già individuato e analizzato i task relativi al problema è risultato banale integrarlo nel progetto.

In particolare i task individuati sono suddivisi in:

- **Parallel Task:** contenenti le fasi di sense e decide degli agenti;
- **Serial Task:** contengono la fase di act degli agenti.

I Parallel Task, data la loro natura, rappresentano task tra loro indipendenti ed essendo quindi compatibili con il framework sono stati assegnati ad un *Executor*. Così facendo è stato possibile rimuovere l'utilità *TaskSplitter* e i relativi *SimulationWorkers* che non sono più risultati necessari.

Inoltre, poiché rimane la necessità di mantenere un punto di sincronizzazione per separare l'esecuzione dei task paralleli da quelli sequenziali, si è scelto di utilizzare un *ExecutorService* e il relativo metodo *submit()*. Questo metodo infatti non solo esegue il task assegnatogli ma gli associa anche un *Future* che attraverso il relativo metodo *get()* permette di ottenere il comportamento desiderato. A questo punto l'utilizzo di una barriera risulta superfluo ed è stata quindi anch'essa rimossa.

L'architettura modificata è riportata in Fig. 1

III. PERFORMANCE

Si è optato per la creazione di una *Cached Thread Pool*, una particolare pool di Thread che è in grado di adattarsi alle necessità del programma, al fine di ottenere delle performance ottimali. Il modello ha delle performance massime leggermente inferiori rispetto all'implementazione fatta con i Thread (circa il 7%), questo dimostra una buona strategia di scheduling utilizzata nel progetto precedente.

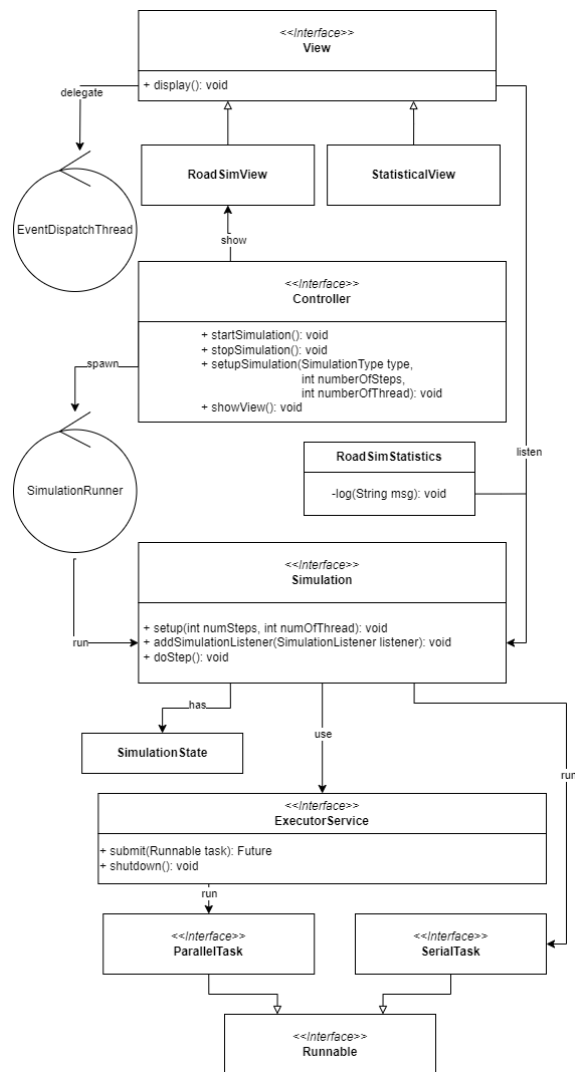


Figure 1. UML