Wirtualne zespoły robocze - instrukcja do zadania z negocjacji agentów w środowisku JADE

Celem zadania jest zapoznanie się z mechanizmami tworzenia agentów w środowisku JADE

Środowisko

Środowisko JADE (ang. Java Agent DEvelopment Framework) jest zestawem narzędzi służących do tworzenia i zarządzania agentami. Zbudowane w oparciu o platformę Java zawiera zestaw klas umożliwiających ponadto komunikację pomiędzy agentami (moduł ACL), rejestrowanie usług w tzw. „żółtej księdze" (agent DF), migrację agentów, definiowanie zachowań, ontologii i wiele innych narzędzi pomocniczych. Podstawowa biblioteka klas wraz z dokumentacją znajduje się na głównej stronie projektu: http://jade.tilab.com/. Program bazowy do ćwiczenia jest zmodyfikowaną wersją programu przykładowego Book-Trading. Pomoc do tego programu znajduje się w pliku JADEProgramming-Tutorial-for-beginners.pdf, można pobrać ze strony projektu lub z lokalnego katalogu C:/JADE/doc/tutorials/.

Informacje praktyczne

Tworzenie projektu w środowisku NetBeans (można też skorzystać ze środowiska Eclipse, IntelliJ IDEA, Visual Studio Code lub używać linii poleceń w terminalu):

- utworzenie projektu typu Java Application bez pliku głównego (należy odhaczyć opcję Create New Class)
- dodanie plików *.java do projektu (wystarczy przeciągnąć pliki do zakładki Source Packages)
- w zakładce Libraries należy umieścić plik jade.jar
- w zakładce Run należy podać nazwę klasy głównej: jade.Boot, oraz parametry uruchomienia np.

```
-agents seller1:BookSellerAgent(); seller2:BookSellerAgent();
buyer1:BookBuyerAgent(Zamek) -gui
```

Uwagi

- Teksty kopiowane z pliku pdf mogą zawierać dodatkowe znaki i przekłamania. Lepiej jest skopiować linię parametrów z nagłówka któregoś z plików źródłowych.

- Interfejs graficzny (GUI) umożliwia zamknięcie programu poprzez wybór odpowiedniej opcji menu. Użycie krzyżyka w systemie Windows zamyka tylko interfejs pozostawiając główny proces. Przed ponownym uruchomieniem programu konieczne jest usunięcie poprzedniego procesu np. poprzez użycie menadżera zadań.

Program bazowy

Program bazowy składa się z dwóch plików, zawierających klasy agentów kupujących i sprzedających przedmioty np. książki. Obie klasy są rozszerzeniami klasy `Agent`. W ramach każdej z tych klas można definiować i dodawać zachowania będące rozszerzeniami klasy `Behaviour` lub pokrewnych : `TickerBehaviour`, `CyclicBehaviour` itd. Po dodaniu zachowania metodą `AddBehaviour()` zaczyna ono funkcjonować podobnie jak niezależny wątek poprzez wykonywanie wewnętrznej metody `action()` w zależności od klasy nadrzędnej. Przykładowo w `CyclicBehaviour` natychmiast po zakończeniu akcji uruchamiana jest kolejna, a w przypadku `TickerBehaviour` akcje są wykonywane w stałych odstępach czasu. Agenty mogą wymieniać się pomiędzy sobą komunikatami. W przypadku, gdy odbiór komunikatów jest możliwy poprzez więcej niż jedno zachowanie (wątek), warto jest zdefiniować szablony komunikatów (używając klasy

`MessageTemplate`), by uniknąć ich odbioru w niewłaściwych miejscach. Przykładowo dwa zachowania są uruchamiane w agencie klasy `BookSellerAgent`.
Kupno-sprzedaż książki w programie bazowym odbywa się w 4 krokach:

- wysłanie zapytania przez kupca o tytuł książki do sprzedawców umieszczonych w wewnętrznym katalogu sprzedawców,
- odbioru zapytania wysłania odpowiedzi przez agentów sprzedających (odmownej gdy książki nie ma w magazynie lub z podaniem jej ceny, jeśli jest dostępna),
- odbioru ofert, wyboru najlepszej i wysłania zamówienia przez kupca,
- odbioru zamówienia i wysłania potwierdzenia jego realizacji przez sprzedawcę.

Po dokonaniu zakupu agent kupiec kończy swoją działalność.

Możliwe zadania do wykonania w ramach tego ćwiczenia

Najprostszym zadaniem jest uzupełnienie klas agentów `BookBuyerAgent.java` oraz `BookSellerAgent.java` w ten sposób, aby agenty typu kupiec i sprzedawca mogły negocjować cenę książki stosując pewne ustalone strategie czyste. Przykładowo kupiec, zamiast od razu złożyć zamówienie po wyborze najlepszej oferty, może zaproponować cenę dwukrotnie niższą, a następnie w kolejnych krokach negocjacji proponować cenę nieco wyższą, obliczaną według jakiegoś wzoru. Podobnie może postępować sprzedawca redukując swoją pierwotnie podaną cenę.

W bardziej złożonym wariancie zadania agenty mogą zmieniać swoje strategie negocjacyjne starając się je optymalizować względem strategii drugiej strony. Należy przy tym uwzględnić nie tylko końcową cenę sprzedaży, ale również koszty negocjacji, które niewątpliwie w praktyce występują, gdy negocjacje przeciągają się lub są zrywane i wymagają szukania innych ofert (czas to pieniądz!). Od strony matematycznej można potraktować takie zmienne strategie negocjacyjne jako grę, zwykle z sumą dodatnią. Taka gra może posiadać punkty równowagi Nasha (gdy nie opłaca się zmienić strategii bez zmiany strategii przez przeciwnika) lub w przeciwnym wypadku może być niestabilna. Zadanie może polegać na zaproponowaniu algorytmu uczenia - adaptacji strategii lub znalezieniu punktów równowagi (lub pokazaniu, że ich nie ma w strategiach czystych). Możliwe jest też zadanie polegające na stosowaniu i dochodzeniu do punktów równowagi w strategiach mieszanych, gdy zmiana strategii polega na zmianie prawdopodobieństwa wyboru strategii czystej spośród kilku zadanych.

Jeszcze inne zadanie może polegać na usprawnieniach technicznych. Przykładowo, zamiast stosowania katalogu sprzedawców, kupiec może używać tzw. żółtej księgi, czyli dynamicznego katalogu sprzedawców realizowanego przez agenta specjalnego DF (Dictionary Facilitator). Dzięki temu, w systemie otwartym, kupiec może korzystać z zawsze aktualnej listy sprzedawców. Sporym technicznym wyzwaniem jest też handel wielu kupców z wieloma sprzedawcami o ograniczonych zasobach.

Zadania w różnych wersjach

Należy uzupełnić dwie proste klasy agentów `BookBuyerAgent.java` oraz `BookSellerAgent.java` w ten sposób, aby agenty typu kupiec i sprzedawca mogły negocjować cenę książki stosując podaną przez prowadzącego wersję strategii negocjacyjnych (4 punkty). Ponadto należy uruchomić usługę katalogową (agent DF, żółta księga) umożliwiającą rejestrację sprzedawców oraz uzyskiwanie ich bieżącej listy po stronie kupca, zamiast statycznego katalogu sprzedawców (1 punkt).

Wersja 1:

- kupiec początkowo odpowiada sprzedawcy ceną o 25% niższą, a następnie zawsze wybiera cenę średnią spośród dwóch ostatnich propozycji,
- sprzedawca zawsze obniża cenę o stałą wartość np. 4, maksymalnie 6 razy.
- kupiec zamawia książkę, gdy cena zaproponowana przez sprzedawcę jest mniejsza lub równa od swojej ostatniej propozycji.

Wersja 2:

- kupiec początkowo odpowiada sprzedawcy ceną o 30% niższą, a następnie zawsze podwyższa cenę o stałą wartość, np. 3, maksymalnie 6 razy,
- sprzedawca zawsze wybiera cenę średnią spośród dwóch ostatnich propozycji (swojej i kupca),
- kupiec zamawia książkę, gdy ostatnia cena sprzedawcy jest taka sama lub mniejsza od ceny ostatnio przez siebie zaproponowanej.

Wersja 3:

- kupiec początkowo odpowiada sprzedawcy ceną o 40% niższą, a następnie zawsze podwyższa cenę o stałą wartość, np. 6, maksymalnie 8 razy,
- sprzedawca zawsze wybiera cenę średnią spośród dwóch ostatnich propozycji (swojej i kupca),
- kupiec zamawia książkę, gdy ostatnia cena sprzedawcy różni się co najwyżej o 3 od ceny ostatnio przez siebie zaproponowanej.

Wersja 4:

- kupiec początkowo odpowiada sprzedawcy ceną o 50% niższą, a następnie zawsze podwyższa cenę o stałą wartość, np. 5, maksymalnie 6 razy,
- sprzedawca zawsze wybiera cenę będącą sumą 3/4 swojej ostatniej propozycji i 1/4 ostatniej oferty kupca,
- kupiec zamawia książkę, gdy ostatnia cena sprzedawcy różni się co najwyżej o 2 od ceny ostatnio przez siebie zaproponowanej.

Wersja 5:

- kupiec początkowo odpowiada sprzedawcy ceną o 20% niższą, a następnie zawsze wybiera cenę będącą sumą 3/4 swojej ostatniej propozycji i 1/4 ostatniej oferty sprzedawcy,
- sprzedawca zawsze obniża cenę o stałą wartość np. 3, maksymalnie 6 razy.
- kupiec zamawia książkę, gdy ostatnia cena sprzedawcy różni się co najwyżej o 2 od ceny ostatnio przez siebie zaproponowanej.

Wersja 6:

- kupiec początkowo odpowiada sprzedawcy ceną o 25% niższą, a następnie zawsze wybiera cenę będącą sumą 3/5 swojej ostatniej propozycji i 2/5 ostatniej oferty sprzedawcy,
- sprzedawca zawsze obniża cenę o stałą wartość np. 2, maksymalnie 5 razy.
- kupiec zamawia książkę, gdy ostatnia cena sprzedawcy różni się co najwyżej o 2 od ceny ostatnio przez siebie zaproponowanej.