Міністерство освіти та науки, молоді та спорту України

Національний університет «Львівська політехніка»

Кафедра САПР

Звіт

до лабораторної роботи №2

на тему:

“Ознайомлення з процесом розроблення мультиагентної системи в середовищі JADE”

з курсу: «Методи проектування мультиагентних систем»

Виконав:

студент групи СПКc-11

Штогрин Олег

Прийняв: Романюк А. Б.

Львів 2014

1. **МЕТА РОБОТИ**

Ознайомитися з процесом розробки мультиагентної системи в середовиці JADE.

1. **КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ**

2.1. Контейнери та платформи

Кожен запущений екземпляр середовища JADE називається контейнером (Container), тому що він може містити декілька агентів. Група (набір) активних контейнерів називається платформою (Platform). Один спеціальний основний контейнер (Main Container) завжди повинен бути активним у платформі і всі інші контейнери ним реєструються як тільки вони створюються. З цього слідує, що перший контейнер, який запускається на платформі повинен бути основним контейнером, а всі інші контейнери повинні бути звичайними “normal” тобто не основними контейнерами і повинні отримати вказівки, де шукати (хост і порт) їх основного контейнера (тобто того контейнера, де вони повинні зареєструватися). Якщо інший основний контейнер запущений де-небудь в мережі, то це буде інша платформа і нові звичайні контейнери можуть бути в ній зареєстровані. JADE-агенти ідентифікуються за унікальними іменами і при умові, що вони знають імена один одного, вони можуть спілкуватися безпосередньо незалежно від їх фактичного місцезнаходження.

2.2. AMS та DF

Крім можливості приймання реєстрацій від інших контейнерів, основний контейнер відрізняється від звичайних контейнерів тим, що він містить два спеціальні агенти (автоматично запускаються коли запускається основний контейнер).

AMS (Agent Management System) – система керування агентами, яка підтримує (забезпечує роботу) службу іменування (тобто, гарантує, що кожен агент всередині платформи має унікальне ім’я) і наділений повноваженнями у платформі (наприклад, може створювати або знищувати агентів у віддалених контейнерах, через запити до AMS).

DF (Directory facilitator) – маршрутизатор каталогів, який підтримує (забезпечує роботу ) службу «Жовті сторінки», за допомогою якої агент може знайти інших агентів, які надають послуги необхідні йому для досягнення своїх цілей.

2.3. Ідентифікатори агента

Кожен агент ідентифікується «ідентифікатором агента», який представляється екземпляром класа jade.core.AID. Метод getAID() класу Agent дозволяє отримати ідентифікатор агента. Об’єкт AID містить глобальне унікальне ім’я та адреси. Ім’я в JADE має форму <nickname>@<platform-name> , якщо агент названий Peter існує на платформі P1 то глобальне унікальне ім’я буде наступне Peter@P1 . Адреси, які входять в AID – адреси платформи де існує агент. Ці адреси використовуються тільки тоді, коли агент має потребу в спілкуванні (комунікації) з іншим агентом, який існує на іншій платформі.

2.4. Завдання агента – клас BEHAVIOUR

Як вже зазначалося, фактична робота, яку повинен виконувати агент, зазвичай, реалізовується через поведінки “behaviours” агента.

Поведінка представляє задачу, яку агент може виконати і реалізовується як об’єкт класу, який успадковується від класу jade.core.behaviours.Behaviour. Для того, щоб агент міг виконати задачу, описану об’єктом поведінки, достатньо додати поведінку агента за допомогою метода addBehaviour() класу Agent. Поведінка може бути додана у будь який момент: коли агент запускається (в методі setup()) або з інших поведінок. Кожен клас, який успадковує Behaviour повинен реалізовувати метод action(), який фактично визначає операції, які будуть виконуватися, коли агент виконує дану поведінку і метод done() (повертає булеве значення), який визначає чи завершена поведінка і чи потрібно її видалити з набору поведінок, які має агент.

1. **ВИКОНАННЯ ЗАВДАННЯ**

В цій лабораторній роботі для ознайомлення з процесом розробки мультиагентної системи в середовищі JADE ми використовували приклад “BookTrading”, який є одним з тестових проектів для ознайомлення з системою JADE. Тому першим етапом виконання цієї лабораторної роботи було забезпечення функціонування цього конкретного прикладу мультиагентної системи. Для цього ми підключили систему JADE до java-проекту, налаштували її та використали код тестового проекту “BookTrading” для розробки мультиагентної системи купівлі/продажу книг.

На рис. 3.1. наведено структуру проекту “BookTrading” в Eclipse IDE:

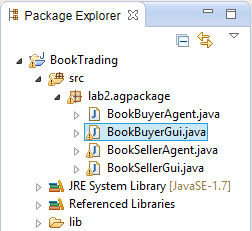


Рис. 3.1. Структура проекту “BookTrading”

Наступним етапом виконання цієї лабораторної роботи було внесення модифікацій в проект “BookTrading”. Я вирішив спочатку розробити графічний інтерфейс покупця для зручнішого використання системи. Вигляд цього інтерфейсу наведено на рис. 3.2.

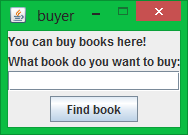


Рис. 3.2. Графічний інтерфейс покупця в системі “BookTrading”

Після цього я створив здійснив такі функціональні зміни:

- створив контейнер, який зберігає каталог книг за допомогою колекції об'єктів "Book" з полями ім'я, ціна та кількість;

- тепер продавець має можливість вносити кількість примірників книги, які є в наявності, і коли покупець купляє книгу, то ця кількість автоматично зменшується на 1. Якщо продавець вносить в каталог книгу, яку вже було додано, то до попередньої кількості примірників цієї книги додається кількість, яку вніс продавець в полі "Amount" в "BookSellerGui".

Зміни в інтерфейсі “BookTrading” після додавання модифікацій наведено на рис. 3.3.-3.4.

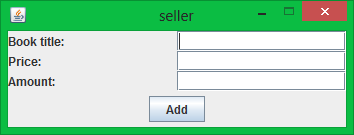


Рис. 3.3. Графічний інтерфейс продавця в системі “BookTrading”

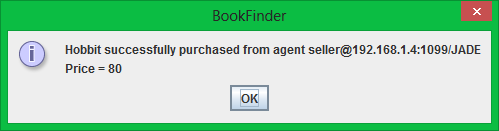


Рис. 3.4. Повідомлення про успішну покупку книги

Також нижче наведено текст інформаційних повідомлень консолі:

Hello! Buyer-agent buyer@192.168.1.4:1099/JADE is ready.

Hobbit inserted into catalogue. Price = 80. Amount = 3

Target book is Hobbit

Trying to buy Hobbit

Found the following seller agents:

seller@192.168.1.4:1099/JADE

Hobbit sold to agent buyer@192.168.1.4:1099/JADE

seller have 2 Hobbit

1. **ВИСНОВКИ**

На цій лабораторній роботі я ознайомився з принципами розробки мультиагентних систем за допомогою системи JADE на прикладі проекту “BookTrading”. Отже, предметна область торгівлі книгами (в даному випадку) складається з покупця та продавця. Саме тому в цьому проекті розроблено два типи агентів “BookBuyerAgent” та “BookSellerAgent”.

Оскільки взаємодія покупця, яка реалізується за допомогою агента-покупця, з даною мультиагентною системою, виконувалась в консольному режимі, я вирішив розробити графічний інтерфейс для покупця. Окрім цього, я додав можливість додавати книги в каталог не по одному примірнику, а одразу задавати кількість цих примірників через графічний інтерфейс продавця.