

Simulare Sistem Bancar: Interacțiunea Agenților ATM și Bancă cu JADE Framework

1. Introducere

Sistemele bancare moderne se bazează pe interacțiuni eficiente și securizate între punctele de acces (cum ar fi ATM-urile) și sistemele centrale ale băncilor. În acest context, proiectul de față își propune să simuleze un ecosistem bancar folosind JADE (Java Agent Development Framework). Scopul principal al acestui proiect este de a dezvolta un microsystem care să includă o Bancă (ca agent central) și mai multe ATM-uri (reprezentate de agenți independenți) ce interacționează între ele prin mesaje ACL (Agent Communication Language).

Proiectul aduce o abordare inovatoare asupra interacțiunii între componentele unui sistem bancar, utilizând conceptul de **sisteme multi-agenți (MAS)** pentru a implementa funcționalități specifice, precum:

- Validarea PIN-urilor utilizatorilor;
- Procesarea cererilor de retragere numerar;
- Verificarea soldurilor conturilor;
- Centralizarea și persistența datelor contabile și a tranzacțiilor.

Această simulare are ca scop oferirea unei perspective reale asupra modului în care pot fi proiectate și implementate soluții distribuite bazate pe agenți, având în vedere cerințele de securitate, scalabilitate și interoperabilitate specifice unui sistem bancar.

2. Obiectivele Proiectului

Crearea unui sistem centralizat pentru gestionarea conturilor bancare:

Agentul Bancă va fi responsabil de stocarea informațiilor despre conturi, validarea PIN-urilor și procesarea tranzacțiilor.

Dezvoltarea mai multor agenți ATM care să comunice cu banca:

Fiecare ATM va fi reprezentat ca un agent autonom, echipat cu o interfață grafică intuitivă pentru utilizator.

Implementarea unui protocol de comunicare securizat:

Se vor utiliza mesaje ACL pentru a transmite cererile și răspunsurile între agenți într-un mod organizat și standardizat.

Persistența datelor:

Asigurarea salvării informațiilor despre conturi și istoricul tranzacțiilor, pentru a permite reluarea sesiunilor fără pierderi de date.

Flexibilitate și scalabilitate:

Sistemul va permite adăugarea dinamică de agenți ATM și închiderea centralizată a tuturor agenților la terminarea sistemului.

3. Arhitectura Sistemului

Sistemul este compus din următoarele componente:

- **Agent Bancă:** Responsabil pentru gestionarea conturilor și tranzacțiilor.
- **Agenți ATM:** Interfața utilizatorului pentru interacțiunea cu sistemul.
- **JADE Platform:** Asigură infrastructura pentru crearea și gestionarea agenților.

Fluxul comunicației

Agentul ATM trimite cereri către agentul Bancă.

Agentul Bancă procesează cererile și trimite răspunsuri.

Operațiunile sunt înregistrate în sistem.

4. Implementarea Sistemului

Codul proiectului este structurat astfel:

§ Pachetul `com.banking.agents`: Conține clasele pentru agentul Bancă și agenții ATM.

§ Pachetul `com.banking.model`: Definește entitățile precum conturi și tranzacții.

§ Pachetul `com.banking.service`: Gestionează persistarea datelor.

Exemple de cod

Următorul cod ilustrează metoda `validatePin` din agentul ATM:

```
public void validatePin(String accountNumber, String pin) {  
    ACLMessage msg = new ACLMessage(ACLMessage.REQUEST);  
    msg.addReceiver(bankAgent);  
    msg.setContent("VALIDATE:" + accountNumber + ":" + pin);  
    send(msg);  
}
```

5. Testare și Depanare

Sistemul a fost testat cu scenarii care includ:

Validarea unui PIN corect și incorect.

Retragerea numerarului cu fonduri suficiente și insuficiente.

Monitorizarea comunicației utilizând Sniffer Agent din JADE GUI.

Probleme și soluții

Eroare: Sistemul nu pornește.

Soluție: Verificați dacă porturile 2099 și 7778 sunt disponibile.

Eroare: ATM nu se conectează la Bancă.

Soluție: Asigurați-vă că agentul Bancă este activ și că rețeaua funcționează corect.

6. Concluzii

Acest proiect demonstrează utilizarea JADE pentru a simula un sistem distribuit bazat pe agenți, cu aplicații practice în domeniul bancar. Sistemul oferă o bază solidă pentru extinderea funcționalităților, precum suport pentru mai mulți agenți sau integrarea cu baze de date mai avansate.