

Metodologii de dezvoltare a sistemelor multi-agent

Manăte Bogdan Ioan

March 6, 2011

Abstract

Următoarea metodologie este propusă pentru dezvoltarea sistemelor multi-agent folosind platforma JADE. Metodologia propusă se axează pe problemele cheie în analiza și proiectarea sistemelor multi-agent. Partea de analiză este generică, iar partea de design se bazează pe exemple concrete bazate pe platforma JADE. Metodologia este ilustrată prin aplicarea ei asupra unor scenarii ipotetice, astfel încât programatorii familiarizați cu platforma JADE pot aplica, analog, această metodologie pentru a reduce semnificativ timpul de dezvoltare al unui nou sistem.

1 Introducere

În dezvoltarea software există patru faze fundamentale: planificare, analiza, design și implementare. În cadrul etapei de implementare este util să se folosească o listă de pași. Această listă de pași reprezintă metodologia de dezvoltare. Folosirea unei metodologii reduce din timpul și efortul necesar, prin cristalizarea pașilor importanți pe care dezvoltatorul trebuie să-i urmărească.

Ingineria produselor software bazate pe agenți, este un câmp relativ nou care poate fi gândit ca o evoluție a programării orientate-obiect. Cu toate că tehnologia orientată pe agenți oferă metode eficiente de rezolvare a anumitor probleme, în momentul de față nu există prea multe metodologii de dezvoltare. Metodologia propusă nu încearcă să extindă tehnicile orientate obiect, în schimb este orientată pe agenți și pe abstractizarea oferită de paradigma orientată pe agenți. Formalizarea fazelor de planificare și implementare din ciclul de viață software nu vor fi prezentate în cadrul acestei metodologii.

Faza de design este orientată pe platforma JADE și asupra conceptelor oferite de aceasta. JADE este un cadru de programare scris în limbajul de programare JAVA, care simplifică implementarea sistemelor multi-agent și care oferă un set de unelte pentru fazele de depanare și lansare. Prin concentrarea specifică asupra platformei JADE în faza de design, dezvoltatorul poate trece peste partea de analiză direct la faza de implementare. Acest lucru va reduce semnificativ timpul necesar dezvoltării aplicației.

2 Prezentare generală a metodologiei

În general o metodologie de dezvoltare software poate fi compusă din:

- Un proces, adică o secvență de pași sau faze care ghidează dezvoltatorul în construirea sistemului.
- Un set de reguli euristice, care ajută dezvoltatorul să facă alegeri relevante.
- Diagrame, scheme sau documente reprezentând unul sau mai multe modele ale sistemului.
- O notare potrivită.
- Un set de șabloane care pot fi aplicate pentru rezolvarea situațiilor comune.
- Unu sau mai multe unelte care automatizează cât mai mult posibil fazele și pașii specificați în proces.

Observând figura 1 se observă că nu există o graniță strictă între faza de analiză și faza de design. În plus metodologia este de natură iterativă, permițându-i dezvoltatorului să se mute între fazele de analiză și design. La sfârșitul fazei de design dezvoltatorul poate să treacă direct la faza de implementare, partea în care intervine scrierea de cod.

În cadrul acestei metodologii se vor face următoarele presupuneri:

- Platforma JADE este aleasă pentru implementare.
- Se vor folosi un număr relativ redus de agenți.
- Structura organizațională este statică, nu se așteaptă un comportament non-emergent în timpul rulării.
- Securitatea nu este importantă.

3 Cum se definește un agent ?

Un agent poate fi văzut ca o aplicație software care rezidă pe o platformă, care îi oferă agentului posibilitatea de a comunica în funcție de complexitatea mediului în care se află.

Agenții pot avea următoarele seturi de proprietăți:

- Autonomie - agenții pot opera fără intervenția utilizatorului, având control asupra acțiunilor lor și stării interne.
- Abilități sociale - agenții pot interacționa cu alți agenți sau cu utilizatorii folosind un limbaj de comunicare.
- Reactivitate - agenții percep mediul în care se află și răspund adecvat schimbărilor survenite.
- Proactivitate - agenții pot lua inițiativă, prezentând comportamente orientate pe îndeplinirea unui anumit scop.

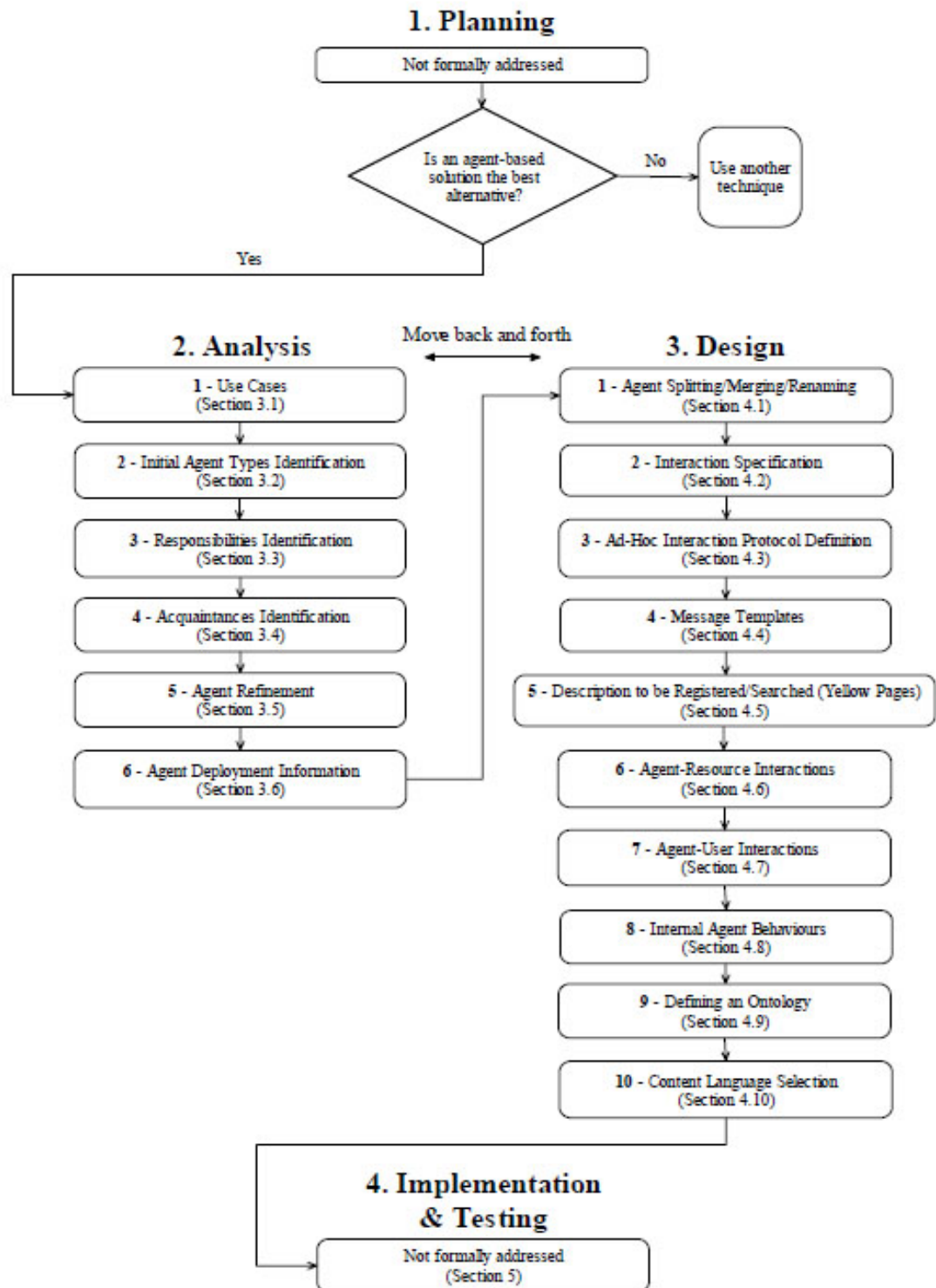


Figure 1: Prezentare generală a metodologiei

4 Scenariu

Scenariul asupra căruia se va aplica această metodologie presupune crearea unei aplicații care să permită organizarea ieșirilor la cinema. Această metodologie va fi de asemenea aplicată analog și în dezvoltarea middleware-ului. Pentru scenariul menționat se presupune că un operator de telefonie mobilă, dorește să le ofere abonaților săi posibilitatea să organizeze evenimente la cinema. Serviciul ar trebui să le permită abonaților să-și invite prietenii să vadă un film, să colecteze preferințele fiecăruia și să ofere cea mai bună opțiune pe baza preferințelor. Sistemul va oferi informații doar despre cinematografele locale în funcție de poziția utilizatorului.

5 Analiza

În partea de analiză se clarifică toate aspectele legate de soluție. În cadrul acestei metodologii, faza de analiză este împărțită în șase pași, definiți în secțiunile următoare.

5.1 Cazuri de utilizare

Cazurile de utilizare reprezintă o metodă efectivă de capturare a potențialelor cerințe funcționale ale unui nou sistem. Fiecare caz de utilizare prezintă unul sau mai multe scenarii care demonstrează cum ar trebui sistemul să reacționeze când interacționează cu utilizatorul sau alt sistem.

Pe baza descrierii aplicației și după chestionarea potențialilor utilizatori este posibil să se construiască o listă preliminară cu scenariile posibile.

5.2 Identificarea tipurilor de agenți

Acest pas presupune identificarea tipurilor inițiale de agenți și construirea unei diagrame de agenți. În această fază se pot aplica următoarele reguli:

- Adăugarea unui tip de agent pentru fiecare utilizator sau dispozitiv.
- Adăugarea unui tip de agent pentru fiecare resursă.

References

- [1] A. H. Bond and L. Gasser (Eds.), Readings in Distributed Artificial Intelligence, Morgan Kaufmann Publishers: San Mateo, CA, 1988.
- [2] S. Bussmann, N. R. Jennings, and M. Wooldridge, Multiagent Systems for Manufacturing Control, Springer-Verlag, 2004.
- [3] G. Caire and D. Cabanillas, “JADE tutorial: creating and using application specific ontologies”, 2004, see: <http://jade.tilab.com/doc/CLOntoSupport.pdf>.

- [4] G. Caire, W. Coulier, F. Garijo, J. Gomez, J. Pavon, F. Leal, P. Chainho, P. Kearney, J. Stark, R. Evans, and P. Massonet, "Agent Oriented Analysis Using Message/UML," *Lecture Notes in Computer Science*, M. Wooldridge, G. Weiss, and P. Ciancarini (Eds.), Springer-Verlag, vol. 2222, 2002, pp. 119-135.
- [5] C. Campo, Directory Facilitator and Service Discovery Agent, FIPA Document Repository, 2002, see: <http://www.fipa.org/docs/input/f-in-00070/f-in-00070.pdf>.
- [6] A. Collinot, A. Drogoul, and P. Benhamou, "Agent oriented design of a soccer robot team," in *Proceedings of the 2nd International Conference on Multi-Agent Systems (ICMAS-96)*, Kyoto, Japan, 1996, pp. 41-47.
- [7] R. Cost, Y. Chen, T. Finin, Y. Labrou, and Y. Peng, "Modeling agent conversations with colored petri nets," in *Workshop on Specifying and Implementing Conversation Policies*, 1999, pp. 59-66.
- [8] A. Dennis and B. H. Wixom, *Systems Analysis and Design: An Applied Approach*, John Wiley and Sons, 2000.
- [9] Foundation for Intelligent Physical Agents (FIPA), see: <http://www.fipa.org/>.