

목표 모델링을 이용한 에이전트 추출 방법

김진태¹⁾ 박수용

서강대학교 컴퓨터학과

jtkim@selab.sogang.ac.kr sypark@ccs.sogang.ac.kr

Agent Selection Method Using Goal Modeling

Jin-Tae Kim¹⁾ Soo-Yong Park

Dept. of Computer Science, Sogang University

요 약

에이전트 분야가 인공지능에서 분리되어 지금까지 각 분야에서 연구되고 있다. 소프트웨어 공학 입장에서 에이전트를 연구할 때, 에이전트를 어떻게 모델링하고 설계하고 구현할 것인가는 가장 큰 연구 이슈가 되어 왔다. 이에 에이전트지향의 소프트웨어 개발 방법에 대한 연구가 진행되었으나, 문제의 영역에서 어떻게 에이전트를 추출할 것인가에 대한 연구는 거의 전무하다고 할 수 있다. 본 논문은 이러한 에이전트 추출에 관한 기준을 정하고 전자상거래를 통한 적용방안을 보여주고 있다. 그리고 각각의 에이전트 속성과 에이전트 내부 구조가 어떻게 매핑 가능한지를 보여 준다.

1. 서론

현재 소프트웨어의 모습은 점차 복잡하고 대형화되어 가고 있으며, 이를 지원하기 위한 소프트웨어 환경 역시 이질적이고 분산된 실시간 환경을 지원하도록 요구되고 있다. 이렇게 시스템이 대형화되고 복잡해짐에 따라, 미래의 소프트웨어 환경에 적합한 하나의 해결책으로 부각되고 있는 에이전트 분야는 초기 인공지능 분야에서 점차 그 응용 분야가 산업부문, 공공부문으로 확대되어가고 있으며, 특히 전자매일을 필터링 하는 에이전트에서부터 비행기 제어(Air-traffic Control) 과 같이 크고 목적이 뚜렷한 복잡한 시스템에 이르기까지 다양하게 사용되고 있다. 또한 전자상거래 응용프로그램들과 같이 분산되고 이질적인 환경에서, 에이전트에 관한 이론들은 널리 받아들여지고 있으며 여러 응용 프로그램의 개발에 적용되고 있다[1].

이러한 에이전트 지향의 소프트웨어들이 개발됨에 따라 소프트웨어 공학적인 측면에서 에이전트 지향의 소프트웨어 개발을 위한 분석적인 접근 방법이 점점 중요해지고 있으며, 체계적인 개발 방법론에 대한 연구가 절실히 필요하게 되었다.

현재 에이전트지향의 소프트웨어 공학 분야에서의 연구는 크게 영역 분석 및 모델링 기술, 에이전트 아키텍처 설계 기술, 에이전트 구현 툴 및 환경 개발 기술의 세 분야로 나눌 수 있다. 영역 분석 및 모델링 기술은 주어진 문제 영역에서 에이전트화 되어야 할 부분을 찾아 모

델링 함으로서 에이전트의 기능, 변화 등을 기술하고 검증하며 소프트웨어 공학의 재사용 개념을 활성화시키는 데 중요한 역할을 하는 기술이다[1]. 그러나, 에이전트 모델링 방법에 대한 연구가 진행되기 전에 문제의 영역에서 어떻게 에이전트를 추출할 것인가는 아직도 많은 어려움으로 남아있다. 따라서 본 논문에서는 문제의 영역에서 에이전트를 추출하는데 있어서, 목표(goal) 지향의 모델링 방법-이하 목표 모델링으로 부른다-을 사용하여 에이전트를 추출하는 방법을 제안하고자 한다.

2. 관련연구

기존에 에이전트 지향의 소프트웨어 개발 방법에는 여러가지가 있다. 그러나 각각의 내용들이 모두 특정 도메인에 대한 내용이기때문에 에이전트 지향의 소프트웨어 개발에 필요한 포괄적인 방법론이 있진 않다. 그러던 중 1999년에 발표된 다음의 논문에선 어느 정도 방법적인 내용을 설명하고 있다.

■ "A Methodology for Agent-Oriented Analysis and Design"

이 논문에서는 에이전트를 분석하고 디자인하는 방법을 제안한다. 그중 에이전트를 추출하는 과정에 사용된 방법은 바로 Role modeling 방법이다. 이 논문에서는 organisation을 collection of roles로 보고 있으며, role model이 시스템에서 key role을 설명한다고 주장한다. Role은 entity의 예상되는 function을 추상적으로

묘사한 것이라고 설명하고 있다

그러나, 이 논문에서 주장하는 Role model의 문제점이 몇가지 있다. 첫째로 시스템의 요구사항은 기능적인 측면 뿐만 아니라 비기능적인 면도 있음에도 불구하고 기능적인 측면만을

보고 모델링을 하고 있다는 점이다. 둘째로 이 논문에서 주장하고 있는 role modeling에 의한 에이전트의 분석 방법에 대한 명확한 기준이 없다. 따라서 기능적, 비기능적인 면을 모두 보여주지 못하며 에이전트 추출 기준이 명확하지 않아 분석가에 따라 많은 이견이 생길 수 있다. 이외의 논문들은 앞에서 언급한 것처럼 각각의 도메인에 맞는 특정 에이전트에 관한 설명이 대부분이며 에이전트를 어떻게 추출하고 분석할 지에 대한 설명은 아직까지 존재하지 않는다.

3. Agent Selection Processes

에이전트 지향의 소프트웨어를 개발하는데 있어서 아직까지 표준화된 방법이 없다[1]. 그것은 에이전트에 대한 개념이 완전히 정립되지 않고 있으며 에이전트가 가지고 있는 3가지 속성-autonomy, adaptation, cooperation-으로써 어떻게 표현해야 할 지에 대한 어려움이 있다. 그러나, 다른 방법론도 그런 것처럼 문제의 시작은 주어진 사용자의 요구사항이다. 바로 그 요구사항을 바탕으로 문제영역에 대한 이해를 가지게 된다. 여기서 특별히 에이전트에게 주어지는 과업을 목표이라고 하는데 이것은 바로 에이전트가 존재하는 이유가 된다. 즉, 에이전트는 주어진 목표를 달성하기 위해 부단히 노력한다.

3.1 전체공정도

본 논문에서는 마치 객체지향 개발 방법론에서 UseCase 다이어그램을 통해 문제를 분석하고 Sequence 다이어그램을 통해 추출된 classe들을 사용하여 class 다이어그램을 만드는 것처럼, goal modeling을 통해 문제의 영역을 분석하고 각각의 leaf node에 대해 goal structure 다이어그램을 완성한다. 각각의 완성된 goal structure 다이어그램을 통해 에이전트지향 규칙을 적용하여 에이전트를 추출하고 최종 결과로 에이전트와 객체가 공존하는 Agent Class 다이어그램이 산출된다. 그림 1은 본 논문에서 제안하는 전체 공정 그림이다.

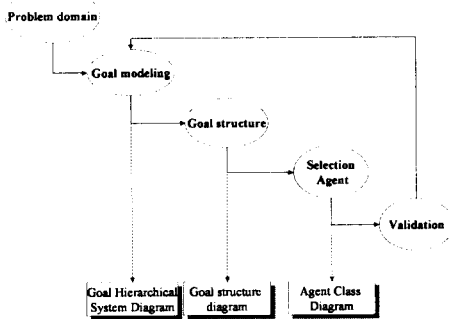


그림 1 전체 공정

3.2 Goal modeling

Goal modeling이란 사용자의 요구사항을 분석할 때에

goal을 중심으로 모델링하는 기법으로 아직까지 완전히 정립된 모델링 방법은 아니다. 지금까지 goal modeling에 대한 주장이 많이 있으며 발전되고 있다. Goal modeling을 통한 산출물은 Goal hierarchical system 다이어그램이 된다

3.3 Goal Structure diagram

Goal modeling을 통해 나타난 Goal hierarchical system 다이어그램에서 각 leaf node에 대하여 goal structure 다이어그램을 생성한다. 그림 2는 goal structure에 대한 그림이다.

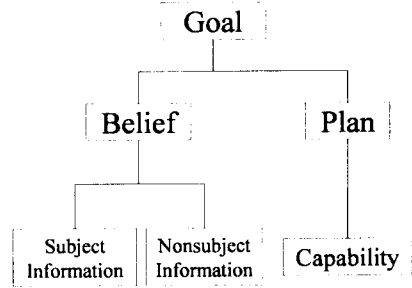


그림 2 goal structure 다이어그램

3.4 Selection of Agent

Goal modeling에 의해 산출된 Goal hierarchical system 다이어그램과 각 leaf node에 대해 생성된 goal structure를 기초로 하여 각 node에 할당된 goal을 성취하는 데에 에이전트로 할 것인지, 객체로 할 것인지 결정하게 된다.

에이전트가 될 것인지 아니면 객체가 될 것인지를 결정하기 위해 먼저 에이전트와 객체를 비교할 필요가 있다.

3.4.1 Agent와 객체

에이전트는 다음과 같은 3가지 속성-autonomy, adaptation, cooperation-을 지닌다.

Autonomy란 에이전트가 사람의 개입 없이 상태(환경)에 의해 독립적으로 자신의 행동을 결정하는 능력을 말한다. Adaptation이란 에이전트가 환경에 위치하여 환경을 지각하고 변화하는 환경에 단순히 반응만 하는 것이 아니라 목적 지향적인 행동을 한다. Cooperation이란 에이전트가 자신의 목표를 이루기 위해 다른 에이전트와도 상호작용하는 능력을 말한다. 따라서 객체와 비교할 때에 에이전트는 능동적이고 활동적인 개체라고 할 수 있다.

3.4.2 Agent Selection Rules

이러한 에이전트의 속성을 나타내기 위해 사용되고 있는 모델이 BDI모델이다. 이는 에이전트의 속성들을 belief, desire, intention으로 구분하여 표현하는 모델을 제공하고 있으며 본 논문에서는 KISS에 발표된 에이전트 지향의 소프트웨어 모델링 방법론에서 제안하고 있는 에이전트 mental state 표현방법을 기반으로 하고 있다. 에이전트의 3가지 속성을 나타내기 위해 goal, belief, plan, capability를 제안하고 있다. 따라서 에이전트의 3가지 속

성이 제안된 모델에 어떻게 나타나는 지에 대한 정의와 각각이 어떻게 agent로 전환 가능한가를 살펴본다.

■ Selection Rule 3

만약 하위 leaf nodes가 agent가 된다면 상위 goal node는 cooperation을 지원하는 coordinator agent가 된다

■ Selection Rule 1

만약 nonsubject가 active, flexible하면, Subject는 Agent가 될 수 있다

Goal structure 다이어그램에 보면 belief 중에서 nonsubject information이 있다. Nonsubject는 주어진 goal에서 plan의 대상이 되는 개체로써 이 nonsubject information의 내용이 능동적이고, 유동적인 성질을 지닌 것이라면 그러한 환경과 상호작용을 하는 주체 또한 적응성의 기능을 가지고 있어야 한다. 따라서 agent로 전환 될 수 있다. 그림 3는 nonsubject의 정보를 가지고 agent를 선택하는 예를 보여준다.

'Identify user's choice'

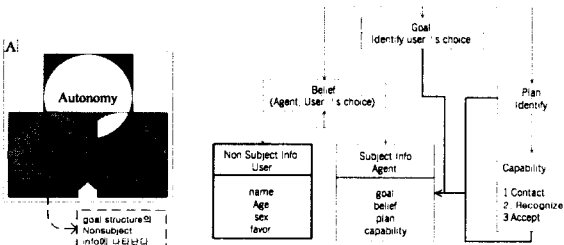


그림 3 Agent의 adaptation과 goal structure의 관계

■ Selection Rule 2

만약 Capability가 searching belief 기능을 갖는다면, Subject는 Agent가 될 수 있다

Goal structure 다이어그램에 나타나는 capability를 살펴보면 capability에 나타나는 내용이 subject info에 대한 belief를 검사하는 부분이 일 경우 agent의 자율성이 반영하게 된다. 이것은 스스로 agent 자신이 누구의 지시없이 자신의 belief를 검색하기 때문에 자율성이 나타나게 된다.

그림 4는 capability에 나타나는 belief 검색으로 agent가 선택됨을 보여준다.

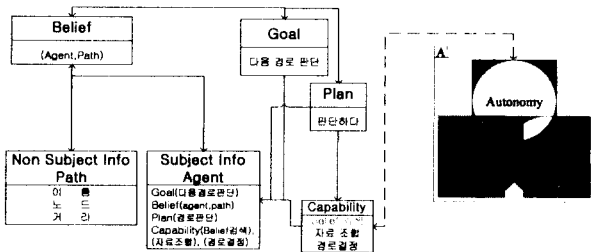


그림 4 Agent의 autonomy와 goal structure의 관계

4. 전자상거래에 적용

위에서 제시한 공정과 결과물을 전자상거래에 적용하면 다음과 같다.

■ Goal hierarchical system 다이어그램

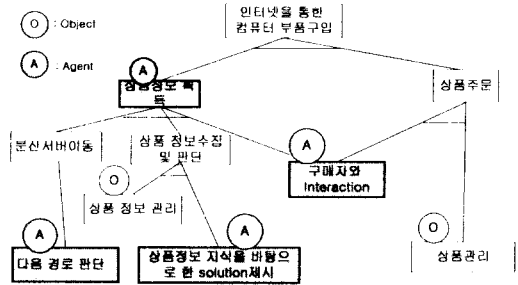
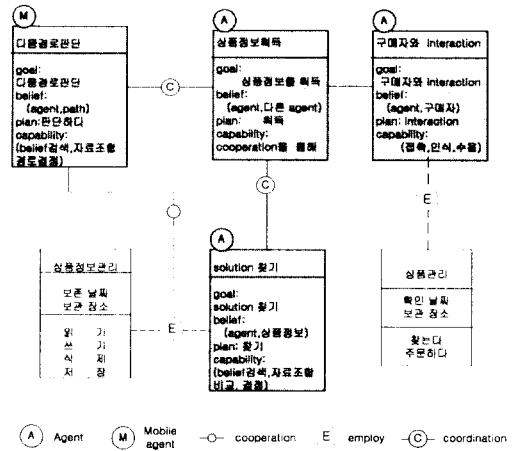


그림 5 Goal hierarchical system 다이어그램

■ Agent Class 다이어그램



○ Agent (A) Mobile agent (M) cooperation (○) employ (E) coordination (C)

그림 6 Agent Class 다이어그램

5. 결론

Agent 지향의 소프트웨어개발에 있어서 중요한 agent 추출 및 분석방법 중 기존의 agent추출, 분석 방법에서 제시하지 못한 기준을 제시하고 있으며, goal 지향의 방법을 사용하여 agent추출과 분석방법을 제시하고 있다. 그러나 향후에 agent추출기준을 정형화하여 나타낼 필요가 있다.

6. 참고문헌

[1] MJ Kim, JT Kim, IJ Park, SY Lee and SY Park, Agent Oriented Software Engineering, APSEC, December 1999.