

# 분산 에이전트 설계 공정 체계화

\*임선종<sup>1</sup>, 송준엽<sup>2</sup>  
<sup>1,2</sup> 한국기계연구원

## Standardization of Design Process for Distributed Agent

\*S. J. Lim<sup>1</sup>, J. Y. Song<sup>2</sup>

<sup>1,2</sup> Intelligent Machine Systems Research Center, KIMM

Key words : Distributed Intelligent Agent, Agent Modeling, Agent Design

### 1. 서론

에이전트는 분산 인공지능 분야에서 문제 해결을 위한 추상적인 단위이며 에이전트 기반의 시스템은 에이전트 정의의 중심 개념으로 하여 시스템을 설계 및 구현하는 것이다.<sup>1</sup> 소프트웨어 개발에서 설계는 요구 분석 사양을 기초로 하여 속성, 성능 등을 실현시킬 수 있는 알고리즘과 그에 의해 처리될 자료 구조의 특성을 명세화하는 것이다. 아직 일치된 견해는 없으나 객체 기반 수준의 프로그램 언어는 설계에 대한 다양한 방법들이 제안되어 사용되고 있다.<sup>2,3</sup> 그러나 에이전트 모델링은 객체의 선정, 내부 구조의 설계, 에이전트간의 관계 설정 그리고 통신 언어 사용 모델 등에서 새로운 개념의 접근 방식으로 공정 설계에 대한 체계화가 미비하다.

본 연구는 에이전트 모델링의 공정 설계 체계화를 목적으로 하고 있으며 이것은 소프트웨어 구조 분석, 성능 예측 그리고 구현을 용이하게 하기 때문에 반드시 필요한 과제이다.

### 2. 에이전트 공정 설계 체계화

소프트웨어 공정 설계를 체계화하는 것은 소프트웨어 개발 생명 주기(Software development life-cycle)의 모든 단계인 계획, 분석, 개발, 설계, 구현 그리고 테스트를 지원하는 것이다. 이것은 올바른 소프트웨어를 개발하기 위한 필수적인 조건이며 가이드가 된다. 에이전트 모델링의 체계적인 설계 공정은 다음과 같다. 에이전트 공정 설계의 기본적인 단계는 Fig. 1 과 같다.

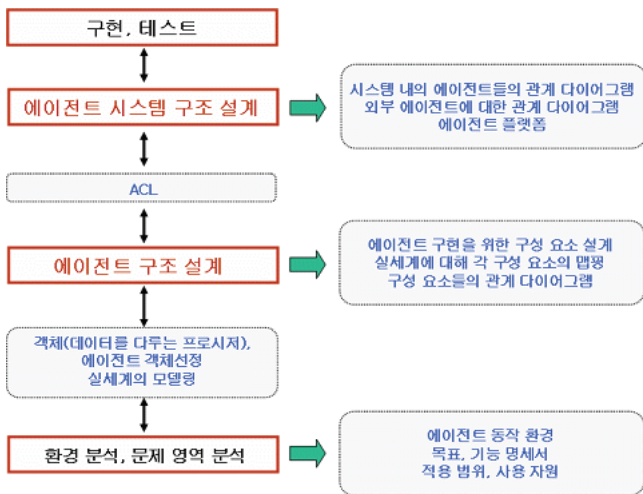


Fig. 1 Development process of agent-based software

#### (1) 에이전트 모델링과 객체 지향 모델링의 비교

객체 기반 모델링과 에이전트 기반 모델링은 정보의 은닉과 상호 작용 모델에서 유사성을 가지고 있지만 몇 가지 차이점을 볼 수 있다. 차이점을 객체의 선정, 분해, 구조 그리고 추상화 측면에서 분석하면 다음과 같다. 객체의 선

정에서 객체 기반의 객체는 데이터와 이를 다루는 프로시저를 기본으로 추출된다. 이 객체는 메시지의 수신에 의해서만 동작하는 수동적인 객체이다. 에이전트 기반의 객체는 에이전트의 기본적인 특징을 포함하도록 선정되며 컨트롤 쓰레드를 가지고 능동적으로 동작한다. 추상화 측면에서 서브시스템을 기술할 때 객체 기반은 데이터와 이의 결과로 표현하지만 에이전트 기반은 목적과 역할로 표현한다. 구조 측면에서 서브 시스템의 관계 설정을 통해 표현되는 복잡한 시스템을 개별적인 행동을 표현하는 객체 기반의 객체보다는 에이전트 특징으로 표현되는 에이전트 기반 객체를 이용하는 것이 적절한 표현을 제공한다.

#### (2) 에이전트의 정의

에이전트의 정의에 대해 Maes, Nwana, Franklin, Graesser, Jennings 그리고 Wooldridge 등이 제안하고 있으며 이 중에서 Jennings 와 Wooldridge 가 제안한 정의를 많이 사용하고 있으며 다음과 같다.<sup>4,5</sup>

“An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives”

#### (3) 환경 및 문제 영역 분석

소프트웨어 설계의 첫 단계로 에이전트 모델링을 적용하게 되는 문제 영역과 적용 환경을 분석하는 것으로 적용 환경, 사용 자원, 적용 범위 등에 대한 내용을 결정한다. 적용 환경으로는 에이전트가 동작되는 네트워크 환경 등에 대한 분석이 필요하다. 사용 자원은 문제 영역을 해결하기 위해 에이전트 시스템이 활용할 수 있는 자원을 의미한다. 적용 범위는 문제 영역에서 에이전트 시스템으로 해결할 수 있는 범위를 의미한다. 문제 영역 분석은 해결이 되어야 하는 문제를 기술하는 것으로 개발의 목적 등을 알 수 있다.

#### (4) 에이전트 선정

에이전트의 기본적인 특징은 반응성, 목표 지향성 그리고 협력성이다. 반응성은 동작되는 환경을 인식하고 변화에 대응하는 기능이다. 목표 지향성은 목적 달성을 위해 창의적인 행동을 할 수 있는 이다. 협력성은 다른 에이전트와 협력할 수 있는 기능이다. 단위 에이전트의 선정 기준은 기본적인 특징의 필요성에 의해 선정된다 특히 반응성과 목표 지향성은 에이전트 선정에 반드시 필요한 항목이다. 시스템에 대한 모델링에서 에이전트 객체의 추출은 기본적인 특징을 필요로 하는 객체로 선정되며 에이전트로 선정될 수 없으나 기능적인 요인이 필요한 것을 수동적인 객체로 분리할 수 있다. 결국 문제 영역은 능동적인 객체와 수동적인 객체의 집합으로 표현된다.

#### (5) 에이전트 구조 설계

에이전트 내부 구조 설계를 위해 필요한 기능은 인식, 동작, 의사 결정, 충돌 관리, 지식 관리, 통신, 학습 그리고 이동 기능으로 볼 수 있다. 의사 결정은 지식을 활용하여 문제 해결을 수행하기 위한 기능이다. 충돌 관리는 에이전트간에 상반되는 결과를 내었을 경우 선택하고 결과에 대해 감시하는 기능이다. 지식 관리는 외부 에이전트와 협력을 위한 속성으로 자신과 외부 에이전트의 ID, 현재 동작

상태, 작업의 적합도 등에 대한 관리 기능을 수행한다. 통신은 다른 에이전트와 메시지를 교환하기 위한 방법이며 적합한 ACL을 선정하게 된다. 학습은 다른 에이전트에 관한 지식을 갱신하는 것이다. 이 단계의 결과물을 요약하면 에이전트 구현을 위한 클래스를 선정하고, 기능 구현을 위한 모듈을 설계하며 모듈간의 관계 다이어그램을 작성한다.

(4) ACL(Agent Communication Language)

FIPA(Foundation of Intelligent Physical Agents)에서는 KQML을 표준으로 사용하고 있다.<sup>6</sup> KQML은 performatives라는 기본 어휘, 매개 변수 그리고 내용으로 구성되어 있다. Performatives는 에이전트간에 이루어지는 행동을 정의하는 기본 어휘이다. Performatives는 basic query, multi-response query, response, generic information, capability-definition 그리고 networking performatives로 구성된다. 의사 소통 모델은 에이전트의 역할에 따라 address routing, brokering 그리고 recommending으로 나눌 수 있다.

(6) 다중 에이전트 구조

에이전트들 사이의 관계를 표현하는 것으로 에이전트간의 정보의 형태, 자율적 행동의 정도, 협력 관계 그리고 문제 해결 능력의 분포에 의존한다. 이 단계의 목적은 지식을 공유하여 목표를 달성하기 위한 것이며 에이전트 역할에 따른 구조, 의사 소통 모델 그리고 에이전트간의 협력 일정, 정도 및 방법 등에 대해 결과를 낸다. Brokering 모델은 facilitator 에이전트의 중재에 의해 임의의 에이전트는 다른 에이전트에 행위를 요청할 수 있다. 에이전트는 자신이 어떠한 행위를 할 수 있음을 facilitator 에이전트에 광고한다. Recommending은 brokering과 유사한 방법이나 facilitator 에이전트는 행위를 수행할 수 있는 에이전트를 소개하는 역할만을 수행한다. Address routing은 임의의 에이전트가 다른 에이전트에게 질의할 수 있다는 것을 알고 있을 경우 ask-performative로 질의하고 tell-performative 메시지로 답을 한다. 다중 에이전트 구조로는 facilitator 구조, 자율적인 에이전트 구조 그리고 계층적인 구조 등을 들 수 있다. Fig. 2, Fig. 3 그리고 Fig. 4는 의사 소통 모델을 보이고 있다.

에이전트의 의사 소통 모델과 구조는 에이전트들의 상호 작용의 형태를 결정하게 된다. 상호 작용의 형태는 다중 에이전트 시스템의 구조와 성능에 영향을 미치게 된다. 상호 작용에 대한 구조는 계층적 구조, federate 구조 그리고 자율적인 구조 등으로 나뉘게 된다. 협력 구조 방식은 mutual adjustment, direct supervision, coordination by standardization, mediated coordination 그리고 coordination by reactive behavior 등으로 구분될 수 있다. Mutual adjustment는 에이전트가 다른 에이전트의 행동에 따라 지식을 공유하는 방식이며 direct supervision은 한 에이전트가 다른 에이전트의 행동을 통제하여 지식의 사용을 제한할 수 있는 방식이다. Coordination by standardization은 감시 에이전트가 다른 에이전트를 조종하는 방식이며 mediated coordination은 mediator가 facilitator로 행동하는 방식이다.

Fig. 2 Brokering model

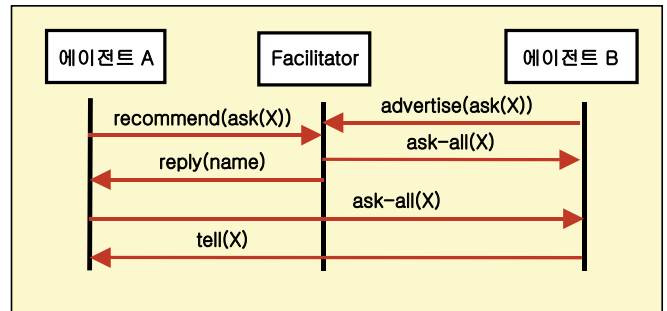


Fig. 3 Recommending model

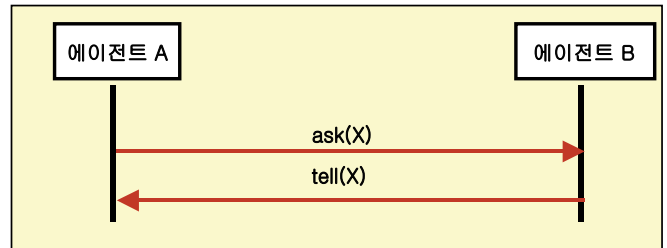


Fig. 4 Address routing

4. 결론

소프트웨어 공정 설계의 체계화는 기능 분석, 성능 예측 그리고 구현을 용이하게 하는 것으로 소프트웨어 개발에 반드시 필요한 단계이다. 본 연구에서 제시하고 있는 공정 설계의 체계화는 구현의 측면에서 에이전트 기반의 모델링에 쉽게 접근할 수 있도록 하고 있다. 공정 설계 체계화를 위해 개발 단계를 환경 및 문제 영역 분석, 에이전트 추출, 에이전트 내부 고조 설계, 에이전트 시스템 구성 설계, ACL 그리고 플랫폼 등으로 구성하였고 각 단계별 결과물의 형태를 제시하고 있다.

참고문헌

1. 이재호, "에이전트 시스템의 연구 및 개발 동향," 정보과학회지, 제 18 권, 제 5 호, pp. 4-9, 2005.
2. 최영근, 허계범, "객체 지향 소프트웨어 공학," 한국실리콘, pp. 33-54, 1997.
3. 유해영, "소프트웨어 공학," 사이텍 미디어, pp. 403-405, 1997.
4. Jennings, R., Wooldridge, M., "Application of Agent Technology. Agent Technology: Foundations, Application and Markets," Springer-Verlag, 1998.
5. Padgham, L., Winikoff, M., "developing intelligent agent systems," WILEY, pp. 1-6, 2004.
6. Finin, T., Weber, J., "Specification of the KQML. Agent-Communication Language," The DARPA Knowledge Sharing Initiative External Interfaces Working Group, 1993.

