

## 다중 작업 처리를 위한 BDI 에이전트 시스템

곽별샘<sup>○</sup> 이재호  
서울시립대학교 인공지능 연구실  
semix2@naver.com<sup>○</sup>, jaeho@uos.ac.kr

### A BDI Agent System Supporting Multiple Parallel Tasks

Byulsaim Kwak<sup>○</sup> Jaeho Lee  
University of Seoul

#### 요 약

소프트웨어는 네트워크로 연결된 다수의 시스템으로 분산되어 주어진 문제를 해결하기 위해 서로 협업하는 방향으로 발전하고 있다. 이러한 시스템의 적절한 예로 멀티 에이전트 시스템을 들 수 있다. 에이전트를 통해 시스템은 느슨하게 결합되어 에이전트의 교체, 추가 및 확장 등이 용이하다. 본 논문에서는 BDI 아키텍처에 기반한 지능형 에이전트 시스템인 VivACE 에이전트 시스템을 제시하고 JADE 플랫폼을 이용해 멀티 에이전트 시스템으로 확장하는 예를 제시한다.

#### 1. 서론

소프트웨어는 네트워크로 연결된 다수의 시스템으로 분산되어 주어진 문제를 해결하기 위해 서로 협업하는 방향으로 발전하고 있다. 이러한 방향은 소프트웨어를 기능별로 모듈화하여 여러 시스템으로 분산시켜서 시스템 자원을 효율적으로 사용할 수 있으며 시스템 유지 보수 및 확장을 용이하게 한다.

이러한 시스템의 적절한 예로 멀티 에이전트 시스템[5]을 들 수 있다. 일반적인 시스템이 서로 다른 시스템과 협업하기 위해 인터페이스를 통해 기능을 호출하는 것과 달리 멀티 에이전트 시스템은 에이전트 통신 언어(Agent Communication Language)를 통해 다른 에이전트와 대화한다. 이것은 기능 기반의 호출이 아닌 작업 기반의 요청으로 인터페이스를 통한 기능 호출보다 추상적인 계층에 속한다. 따라서 시스템은 느슨하게 결합되고 에이전트의 교체, 추가 및 확장 등이 용이하다.

에이전트는 주어진 문제를 해결하기 위해 그것이 놓인 환경을 인식하고 판단하여 스스로 적절한 행위를 결정한다. 특히 지능형 에이전트는 동적인 환경의 변화를 민감하게 인식하고 적절히 판단하여 적응하면서 주어진 목적을 달성한다.

본 논문에서는 BDI 아키텍처에 기반한 지능형 에이전트인 VivACE 에이전트 시스템을 제시하고 JADE 플랫폼[3]을 이용해 멀티 에이전트 시스템으로의 확장하는 예를 제시한다.

#### 2. VivACE 에이전트 시스템

본 논문이 제시하는 VivACE(Vivid Agent Computing Environment) 에이전트 시스템은 BDI 아키텍처에 기반을 두고 있다. BDI 아키텍처는 다음과 같이 구성된다[1][4].

- 믿음(Belief): 에이전트는 그것이 놓인 환경으로부터 인식한 결과에 대해 내부적인 믿음을 생성하며 이것은 행위를 결정하기 위한 판단의 근거로서 활용된다.
- 목적(Desire): 에이전트가 수행해야 할 문제를 정의한다.
- 의도(Intention): 에이전트가 주어진 목적을 달성하기 위해 환경으로부터 인식된 믿음으로부터 판단한 결과로 선택된 행위가 수행되고 있을 때 이 행위를 에이전트의 의도라고 한다.

이에 기반하여 시스템은 다음과 같이 구성된다.(그림 1) 시스템의 가장 중심에 있는 실세계 모델(World Model)이 에이전트의 믿음에 해당한다. VivACE 에이전트 시스템은 모든 판단의 근거를 실세계 모델에서 찾는다. 실세계 모델에는 인식기(Sensor)를 통해 에이전트 외부로부터 인식된 환경 정보와 에이전트가 수행 중에 생성해 낸 에이전트 내부 정보 등 행위를 결정하고 판단하기 위해 필요한 모든 정보들을 포함한다. 이러한 정보들은 관계(Relation)라 하여 기호적으로 표현된다. 관계는 그것을 의미하는 이름과

관련된 요소들로 구성된다.

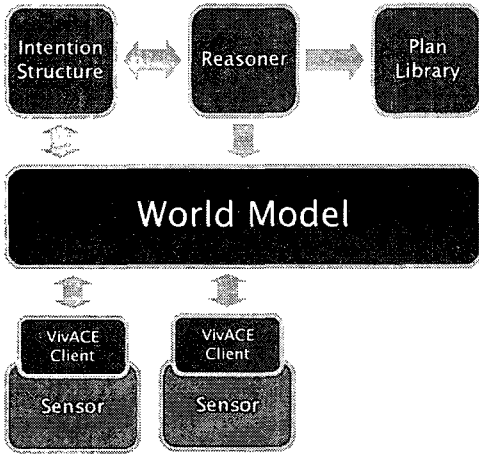


그림 1. VivACE 에이전트 시스템의 구조

에이전트가 외부의 환경을 인식하기 위해 인식기 (Sensor)를 추가할 수 있다. 인식기는 에이전트 외부의 환경을 인식하고 그것을 에이전트가 활용할 수 있는 적절한 정보로 가공하여 실세계 모델에 반영한다. 인식기는 에이전트의 실세계 모델에 직접 접근하지 않고 VivACE 클라이언트를 통해 접근한다.

VivACE 클라이언트는 실세계 모델에 대한 접근 권한을 부여하고 권한에 따라 실세계 모델에 관계(Relation)를 추가, 제거, 변경, 질의 할 수 있는 오퍼레이션을 제공한다. 이와 함께 새로운 목적을 생성할 수 있는 오퍼레이션을 제공하는데 이는 실세계 모델에 반영할 새로운 정보에 대해 특수한 처리가 필요한 경우에 유용하다.

VivACE 에이전트 시스템은 그것의 목적(Goal)을 달성하기 위한 세세한 행위의 조합 전체를 추론해내지는 않는다. 대신 특정 목적을 달성할 수 있는 일련의 행위를 미리 정의하며 이것을 실행 계획(Plan)이라 한다. 실행 계획 라이브러리(Plan Library)는 이렇게 미리 정의된 실행 계획들의 집합이다. 실행 계획에는 다음과 같은 내용이 서술된다.

- 달성할 수 있는 목적
- 목적이 달성되기 위해 만족되어야 하는 조건
- 실행 계획을 수행하기 위해 필요한 선행 조건
- 실행 계획이 실행되는 동안 유지되어야 하는 조건
- 목적을 달성할 수 있는 행위의 집합

실행 계획에 서술된 행위에는 새로운 목적을 생성하거나 하위 목적을 달성하는 것이 포함된다. 따라서 일련의 행위를 미리 정의하는 것이 에이전트의 행위를 고정되는 것을 의미하지 않는다. 그것이 생성하는 하위 목적에 대해 상황에 적절한 실행 계획이 선택되고 수행됨으로써 행위

의 다양한 조합이 이루어진다.

판단기(Reasoner)는 달성하려는 목적에 대해 실행 계획 라이브러리로부터 목적을 달성할 수 있는 실행 계획들을 검색하고 실세계 모델에 표현된 현재의 상황을 고려하여 적절한 실행 계획을 선택한다. 달성하려는 목적과 그것을 달성하기 위해 실행 중인 실행 계획을 의도 구조(Intention Structure)라 한다. 앞서 설명했듯이 실행 계획이 수행되는 동안 하위 목적을 생성할 수 있고 그렇게 생성된 하위 목적에 대해 적절한 실행 계획이 선택되므로 의도 구조는 다음과 같은 모습을 한다(그림 2).

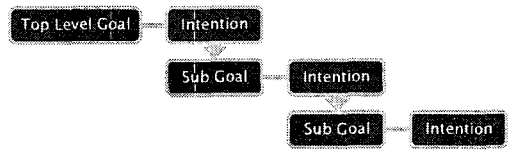


그림 2. 의도 구조

실행 계획이 실행 중에 생성한 하위 목적은 그것을 파생시킨 상위 목적의 실행 계획에 서술된 제약 조건들을 모두 포함한다. 따라서 하위 목적을 달성하고 있는 상황에서 상위 실행 계획에 서술된 유지 조건을 위배하게 되면 그것을 포함한 모든 하위 실행 계획들을 실패로 간주하고 수행을 종료하며 그 시점에서 다시 판단 과정을 통해 현재 상황에 적절한 실행 계획을 검색하여 실행한다.

VivACE 에이전트 시스템은 동시에 다수의 목적을 수행할 수 있으며 동시에 수행되는 목적들은 의도 구조 내에서 최상위 목적들로 존재한다.

### 3. 멀티 에이전트 시스템으로의 확장

멀티 에이전트 시스템이 효과적으로 운용되기 위해서는 에이전트의 등록 및 관리, 검색, 통신을 지원하는 에이전트 미들웨어가 필요하다. VivACE 에이전트는 이러한 조건을 만족하는 잘 알려진 JADE 플랫폼[3]을 활용하여 멀티 에이전트 시스템 환경을 구축한다(그림 3).

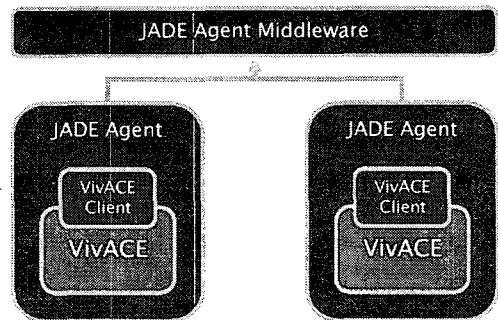


그림 3. 구축된 멀티 에이전트 시스템 환경

VivACE 에이전트는 JADE 플랫폼이 제공하는 다양한 기능을 그대로 사용한다. 특히 ACL(Agent Communication Language) 메시지[1]를 통한 에이전트 간의 통신 기능을 적극 활용하여 협업 체계를 구축한다.

VivACE 에이전트는 모든 행위가 목적에 기반하므로 ACL 메시지를 주고 받는 것 또한 목적과 행위로 구분된다. 구축한 멀티 에이전트 시스템 환경에서는 ACL 메시지를 보낼 수 있는 기본적인 실행 계획이 포함되어 있다. 특정 목적을 수행하는 실행 계획이 다른 에이전트에게 ACL 메시지를 보내야 하는 경우 ACL 메시지를 보내도록 하위 목적을 생성하면 생성된 목적에 의해 기본적으로 포함되어 있는 실행 계획이 선택되어 그것을 통해 ACL 메시지를 다른 에이전트에게 보낼 수 있게 된다.

다른 에이전트로부터 도착한 ACL 메시지는 VivACE 클라이언트로부터 인식된다. 구축한 환경에는 ACL 메시지를 전달하는 VivACE 클라이언트가 포함되며 이것은 새로운 ACL 메시지가 도착하면 그것을 실세계 모델에 반영하고 적절한 처리를 하도록 최상위 목적을 생성한다. 메시지의 특징에 따른 다양한 실행 계획들 중에서 판단기는 적절한 실행 계획을 선택하고 실행한다.

구축한 멀티 에이전트 시스템 환경은 JADE 플랫폼을 그대로 활용하기 때문에 VivACE 에이전트가 아닌 JADE 에이전트를 구현한 다른 에이전트들과도 협업 체계를 구축할 수 있다.

#### 4. 결론 및 향후 계획

VivACE 에이전트는 주어진 목적에 의해 행위가 결정된다. 실행 계획은 특정 목적을 달성할 수 있는 일련의 행위와 그것을 실행하기 위해 필요한 선행 조건, 실행을 유지하기 위해 필요한 유지 조건 등을 서술하여 상황을 고려한, 상황에 특화된 일련의 행위를 기술할 수 있다. 따라서 하나의 목적에 대해 모든 상황을 고려한 하나의 실행 계획이 존재하는 것이 아니라 각 상황에 적절한 다수의 실행 계획이 존재한다. VivACE 에이전트는 주어진 목적을 달성할 수 있는 실행 계획들 중에서 현재 상황을 고려하여 가장 적합한 실행 계획을 수행하여 주어진 목적을 달성한다.

또한 행위가 새로운 목적을 생성하거나 하위 목적을 달성하게 함으로써 소수의 실행 계획으로부터 매우 다양한 행위의 조합을 만들어낼 수 있다. 수행 중인 실행 계획은 그것이 서술한 유지 조건을 만족하지 못하면 안전하게 종료되고 상황에 적합한 다른 실행 계획을 선택하여 목적을 달성하므로 환경의 변화에 민감하게 반응하고 적응하여 최종적으로 주어진 목적을 달성할 수 있게 된다.

멀티 에이전트 시스템으로의 확장은 각각의 VivACE 에이전트 시스템을 특정 환경과 목적에 특화하고 서로 협업하게 함으로써 주어진 문제를 보다 효율적으로 해결할 수 있게 할 뿐만 아니라 시스템을 매우 유연하게 구성할 수 있게 한다. VivACE 에이전트를 이용한 멀티 에이전트 시스템은 유비쿼터스 환경처럼 네트워크로 분산되고 환경의

다양한 변화에 대해 적절한 적응을 필요로 하는 환경에 매우 적합하다.

VivACE 에이전트 시스템은 주어진 목적들을 동시에 추구할 수 있지만 아직 목적들이 서로 어떤 영향을 미치는가에 대한 서술, 즉 목적들간의 온톨로지[6]를 표현하지 못한다. 목적들간의 온톨로지를 서술할 수 있게 되면 실행 계획에 서술해야 했던 많은 상황 조건들이 목적들간의 온톨로지로 해결될 수 있을 것이다. 더불어 실세계 모델에 표현된 정보들에 대해서도 그것의 의미와 관계를 표현하면 보다 지능적인 행위를 취할 수 있을 것이다. 실세계 모델에 표현된 정보의 의미를 파악하고 그것들간의 관계를 통해 새로운 사실을 추론할 수 있다면 실행 계획에 서술하는 다양한 조건들을 추론 가능한 적절한 상황으로 서술할 수 있다. 이를 통해 실행 계획을 서술하는 다양한 조건들과 그것이 표현하는 상황에 대한 의미를 시스템 외부로 뽑아냄으로써 실행 계획을 에이전트가 놓인 환경으로부터 독립적으로 설계할 수 있고, 따라서 다른 환경에 놓인 에이전트라 할지라도 그것이 서술한 온톨로지를 통해 실행 계획을 환경에 맞게 적절히 재사용할 수 있을 것이다.

#### 5. 후기

본 논문은 과학기술부 특장연구개발사업 과제인 "자동차 모듈 설계용 e-엔지니어링 프레임워크 개발"의 일부이며 연구수행에 지원해 주신 관계자 여러분께 감사 드립니다.

#### 6. 참고 문헌

- [1] FIPA Agent Communication Language Specifications. <http://fipa.org/repository/acispecs.html>
- [2] Jaeho Lee, Edmund H. Durfee, "Structured circuit semantics for reactive plan execution systems," *Proceedings of the twelfth national conference on Artificial intelligence* (vol. 2) p.1232-1237, October 1994.
- [3] Java Agent Development Framework. <http://jade.tilab.com/>
- [4] Marcus Huber, "JAM, A BDI-theoretic Mobile Agent," *Proceedings of the Third International Conference on Autonomous Agents (Agents-99)*, May, 1999.
- [5] Michael Beetz, "Autonomous Agents and Multi-Agent Systems," Volume 4, Issue 1-2, p.25-55, March-June 2001.
- [6] Mike Dean, et al, "OWL Web Ontology Language Reference," <http://www.w3.org/TR/owl-ref/>