

대화하는 에이전트(Conversational Agents)

Web을 위한 에이전트를 구축하거나 구입할 때, 우리는 그 에이전트가 가능한 한 잘 수행하기를 바란다. 이는 점점 에이전트가 Web의 정보 자원들 뿐만 아니라 거기에서 운영되는 다른 에이전트들을 이용할 수 있어야 함을 의미한다.

그리고 멀지 않아 대단히 많은 에이전트들이 존재할 것이다. 정보적으로 혼란한 Web으로부터 진화할 것으로 보이는 계산 구조는 사용자, 서비스, 그리고 데이터 자원들을 대표하는 무수한 에이전트들로 구성된다. 이러한 구조에서의 전형적인 사용 패턴으로, 자원 에이전트는 서비스들에 광고하고, 사용자 에이전트는 이러한 서비스를 이용하여 자원 에이전트를 찾아 필요한 정보에 대해 질의하는 것이다.

서로 다른 사용자들을 대표하는 에이전트들이 정보를 찾고 융합하는데 협력할 수도 있으나, 상품과 자원을 위해 경쟁할 수도 있다. 유사하게, 서비스 에이전트들은 사용자, 자원, 그리고 다른 서비스 에이전트들과 협력하거나 경쟁할 수 있다. 그들이 협력자이건 경쟁자이건 간에, 에이전트들은 목적을 갖고 상호작용해야만 한다. 대부분의 의도적인 상호작용은 - 정보를 알려주거나, 질의하거나, 또는 속이기 위하여 - 에이전트들이 서로 대화하는 것을 필요로 한다. 그리고 이해 가능한 대화를 위해서는 서로가 이해할 수 있는 언어가 필요하다.

◎ 공통어(Lingua Franca)

에이전트 프로젝트들은 여러 해 동안 언어를 연구하였다. 초기에는 에이전트들이 각 프로젝트들에 국한되었고 그들의 언어들은 각자 특유한 방법으로 정의되었으나, 이제 에이전트가 단지 사용자뿐만 아니라 다른 에이전트와 대화할 수 있게 하는 것이 필요하게 되었다. 이를 위한 해결책은 공통어이다. 이상적으로, (동일한) 공통어를 사용하는 모든 에이전트들은 상호간에 이해 가능할 것이다.

이러한 공통어는 어떤 언어이어야 할 것인가? 이

언어는 명확한 구문을 가져 모든 에이전트들이 같은 방법으로 구문 분석할 수 있어야 한다. 또한, 이 언어는 잘 정의된 의미를 가져야 한다. 그래서 에이전트들이 동일한 방식으로 문장을 이해할 수 있어야 한다. 이러한 언어의 구문과 의미는 널리 공개되어 여러 설계자들이 그것을 구현할 수 있게 하고, 결과적으로 이렇게 구현된 많은 에이전트들이 서로 대화할 수 있어야 한다. 이러한 언어는 또한 에이전트들이 상호 원하는 것을 충분히 표현할 수 있는 표현력을 갖추어야 하는데, 이것은 우리가 짐작할 수 있듯이 간단한 문제가 아니다. 언어의 명확한 구문을 제안하는 것은 가장 쉬운 일이다. 언어가 널리 알려지는 것은 언어에 대한 기술적이라기 보다는 정치적인 요구조건이다. 위원회 또는 컨소시엄 등을 구성하여 언어가 광범위하게 채택되게 노력할 수 있다. 언어의 표현력을 보장하는 것은 근본적으로 매우 어렵다. 하지만, 인간 언어를 연구하여 많은 훌륭한 아이디어들을 차용할 수 있다.

그러한 의미에 대한 질문은 아직 남아 있다. 아직 아무도 그것을 명확히 밝히지 못했으며, 그 이유는 곧 설명하겠다.

◎ 현재 선택할 수 있는 언어

그러면 에이전트가 이해하고 이해될 수 있도록 에이전트에게 어떤 언어를 제공해야(또는 가르쳐야- 이것은 미래의 주제이지만) 하는가? 현재 두개의 강력한 후보가 있다: 즉 KQML(Knowledge Query and Manipulation Language)*과 FIPA(Foundation for Intelligent Physical Agents)*이다. 두 언어 모두 의사소통의 도메인에 종속적인 부분 내용을 도메인에 독립적인 부분 패키징으로부터 분리하고, 그리고 나서 도메인에 독립적인 부분을 위한 표준을 제공하고 있다.

두 언어 모두 "언어 행위(speech acts)"에 기반을 두고 있다(Box 안의 보충 해설 참조). 비록 KQML이

더 일찍 시작되었고 현재 더 성숙하기는 하지만 둘 다 아직 완전하지는 않다. 두 언어 모두 형식화(formalize)되고 있고, 따라서 이 표준을 따르도록 개발된 에이전트들은 서로 이해할 수 있을 것이다. 두 언어 모두 프로토타입이 구현되었고, 초기 응용 분야를 갖고 있으며, 지지자들이 있다.

◎ KQML

KQML(Knowledge Query and Manipulation Language)은 DARPA가 지원하는 Knowledge Sharing Effort 1 프로젝트에서 정의되었다. KQML은 최하위 층에 메시지 전달 또는 통신을 위한 기능을 가지며, 최상위 층에 응용프로그램들에 의해 지정되는 내용 일반적으로 KIF(Knowledge Interchange Format)* 또는 데이터베이스를 위한 SQL(Structured Query Language)과 같은 형식 언어로 지정을 갖는 계층 구조를

◎ 언어행위(Speech Acts)

언어 행위는 의사소통(communication)과 관련이 있다. 인간의 의사소통이 종종 언어를 포함하는 것을 제외하고는 언어 자체와는 아무런 관련이 없다.

언어 행위 이론은 50년대와 60년대에 인간의 언어를 이해하는 것을 돕기 위해 고안되었다. 중요한 아이디어는 우리가 언어를 사용하여 문장을 만들 뿐만 아니라 행위도 수행한다는 것이었다.⁽¹⁾ 예를 들어, 우리가 뭔가를 요청할 때, 우리는 이러한 요청을 단지 기술하는 것이 아니라 실제로 요청이 발생하게 하는 것이다. 판사가 한쌍의 남자와 여자를 부부로 선언할 때, 부부상태에 관해 보고만 하는 것이 아니라 그 상태를 변경하고 있는 것이다.

“나는 ……을 요청한다” 또는 “나는 ……을 선언한다”로 시작하는 언어 행위의 양식화된 구문 형식을 수행문(performative)이라 한다. 말 그대로, 수행문으로 말을 하므로써 그렇게 만드는 것이다. 이 형식에 사용될 수 없는 동사들은 언어 행위가 아니다. 예를 들어, “해결하다(solve)”는 수행문이 아니다. 왜냐하면, “나는 이 문제를 해결한다(I hereby solve this problem)”로 말한다고 문제가 해결되지는 않기 때문이다. 그렇게 된다면 수학을 공부하는 학생들이 훨씬 더 행복해질 것이다.

영어에서 수백 개의 동사들이 수행문에 해당된다. 이를 분석하기 위해 분류가 필요하게 되었고, 현재 많은 연구가 되어있다. 대부분의 전산 목적을 위하여, 언어 행위들은 다음과 같이 분류된다.

- 주장(assertive)(알려주는)
- 지시(directive)(요청 또는 질의하는)
- 공약(commisive)(약속하는)
- 금지(prohibitive)
- 선언(declarative)(예를 들어, 판사가 결혼식에서 행하는 것과 같이 그 자체에서 사건을 야기하는)
- 표현(expressive)(감정을 표현하는)

자연어에서 어떤 언어 행위가 수행되고 있는가를 결정하는 것은 쉽지 않다. 예를 들어, 만약 A가 “여기는 춥다”라고 말한다면 그는 온도에 관해 말하고 있을 수도 있고, 또는 히터를 가동하길 원하고 있을 수도 있다. 이것은 자연어가 미묘한 이유들 중의 하나이다.

인공어에는 이러한 문제점은 없다. 언어 행위들의 의미는 에이전트들이 수행방법에 대하여 믿고 의도하고 있는 것과 그들이 살고 있는 사회에 따라 달라질 수 있다. 이러한 모든 관련된 것들이 미묘하기 때문에 의미의 특성을 분석하는 일이 어려운 것이다.

참고문헌

1. J.L. Austin, *How to Do Things with Words*, Clarendon Press, Oxford, UK, 1962.

갖고 있다. 이러한 최상위층과 최하위층 사이에는 에이전트들이 의미있는 메시지를 교환하기에 필요한 기본 요소들이 존재한다. 기본적으로 KQML은 이러한 메시지들을 구성하는 방법을 제공하지만, 구체적인 내용은 에이전트 설계자들에게 필요한 사항들이기 때문에 여기서는 생략한다.

보충 해설의 전문용어로 말하면, KQML은 31과 41 사이의 수행문(performative)을 제공한다고 할 수 있다. 변형된 형태이지만, KQML의 수행문들은 모두 주장과 선언들이다. 이들은 몇개의 주요 클래스로 나뉘어진다. 첫번째 클래스 "tell", "evaluate", 그리고 "subscribe"를 포함하는 내용의 실제적인 소통에 맞추어졌다. 두 번째 클래스는 정보의 흐름을 제어하기 위한 기본요소들을 포함한다. 예를 들어, "next"를 보냄으로서 다음 답을 원하는 것이다. 세 번째 클래스는 새로운 에이전트들의 모집, 다른 중개업의 수행, 그리고 기능의 촉진 등을 포함한다.

KQML은 메시지 전달은 잘 되었다는 가정 하에 메시지의 순서를 보존한다. 그러나, 전달 시간은 보장하지 않는다. 이러한 이유로, 통신의 기본 패러다임은 비동기이다. 응용 층에서는, 메세지들의 관계를 정의하는 꼬리표를 이용하여 동기 통신의 효과를 얻을 수 있다. 예를 들어, 응답을 해당 질의에 연결시킬 수 있다. 이러한 방식으로 KQML은 기본적인 상호작용 프로토콜들을 지원한다. 하지만 보다 복잡한 프로토콜들은 KQML 외부에서 구현되어야 한다.

KQML의 의미론은 비형식적으로 주어지며, 형식화 작업은 현재 진행 중이다. KQML 에이전트들은 믿음(beliefs)과 목적(goals)을 포함하는 가상 지식기반(virtual knowledge base, VKB)을 갖는다고 가정된다. 그들은 그들 자신의 가상 지식기반과 다른 에이전트들의 가상 지식기반에 대하여 의사소통을 할 수 있다. 예를 들어, "tell"은 송신자의 VKB 내용을 보고하고, "evaluate"는 수신자에게 그의 VKB에 대해 보고하도록 지시한다.

◎ FIPA

유럽과 아시아, 그리고 미국 회원들로 구성된 컨소시엄인 FIPA(Foundation for Intelligent Physical Agents)는 다른 언어행위기반(speech-act-based) 언어를 제안하였다. FIPA는 KQML보다 작은 단지 6개로 이루어진 수행문을 가지고 있지만, 이것들은 에

이전트들의 더 복잡한 믿음(beliefs)과 기대(expectations)를 표현하기 위해 복잡할 수 있다. 예를 들어, 에이전트는 여러 대상 중 하나를 알기 위하여 요청할 수 있다. 수행문은 행위를 명시적으로 처리하므로, 전달된 행위를 메시지 수신자가 수행하기를 요청한 것이다.

FIPA 규정은 형식적인 의미론을 갖추고 있다. 이는 일반적으로 장점이 된다. 예를 들어, 이것은 한 에이전트의 메시지가 한가지 방법으로 해석됨을 보장한다. 이러한 보장이 없다면, 에이전트들은(그리고 그 설계자들은) 오해의 소지가 있고 여러 대안들 중에서 하나를 선택해야만 하는데 이는 불필요한 작업이 될 수 있다.

그러나, FIPA의 의미론은 어떤 특정한 방식으로 행동하는 특정한 종류의 에이전트에 한정되어 있다. 에이전트들은 진실해야 하며 그들의 믿음에 절대적인 확신이 없으면 주장(assertions)을 만들지 말아야 한다. 예를 들어, FIPA는 에이전트들이 다른 에이전트에게 통보하는 시점에 대해 제한을 두었다(즉, 에이전트들이 뭔가를 알고 있지만 상대방은 그렇지 않을 경우). 이러한 방법으로 불필요한 메세지 살포를 방지할 수 있는데, 그것은 발송 에이전트가 모든 수신 에이전트가 메세지 내용을 모른다고 확신할 수 없기 때문이다. 나아가 엔지니어 에이전트는 가능성 있는 설계 방안들에 대해 논의할 수 없고, 정치가 에이전트는 청중들이 그가 하고있는 말을 믿고 있을 경우에만 계속하여 주장할 수 있을 것이다! 이것은 에이전트들이 살아가는 사회를 지나치게 법제화하는 것이다. 우리는 무엇이 에이전트들의 올바른 행동인가 알고 있다 에이전트의 행동을 결정하는 것은 프로토콜의 기능은 아니다.

◎ 평가

그러면 어느것을 선택하는 것이 좋을까? KQML은 여전히 불충분한 의미론이 약점이다. 결과적으로, KQML 구현들은 제각각인 것처럼 보인다. 이것은 상호 의사소통을 어렵게 만들어서, 당신의 KQML 에이전트를 아무도 이해하지 못할 수도 있다. 또한 KQML에는 보안에 대한 언급이 없다. 에이전트의 신원을 확인하거나 KQML 메시지의 무결성을 보장하기 위한 아무런 규정이 마련되어 있지 않다.

이와는 대조적으로 FIPA 규정은 의미론을 형식

참고문헌

화하고 보안 모델을 제공하기 위해 시도하였다. 그러나, 새로운 규격인 관계로 아직 광범위하게 시험되거나 채택되지 않고 있다. 결과적으로 FIPA 에이전트는 통신할 수 있는 상대를 발견하지 못할 수도 있다.

두 언어의 중요한 의미상 차이는 다음과 같이 설명될 수 있다. FIPA에서는 에이전트 A가 에이전트 B에게 무언가 말할때 이는 B가 그것을 믿기를 A가 바라는 것이고, KQML에서는 에이전트 A가 에이전트 B에게 무언가 말할때 이는 A는 자신이 그것을 믿고 있다는 사실을 B가 알아주길 바란다는 것이다. 이런 의미에서 KQML이 좀 더 신중하다고 할 수 있다.

만약 당신이 사용자들간의 합의가 이루어져 한 언어가 산업 표준으로 결정될때까지 기다릴 수 없다면, KQML을 선택할 것을 권한다. 왜냐하면 현재 시장 점유율이 가장 높기 때문이다. 그 다음, KQML의 의미론 표준화 노력에 희망을 걸고 기다리는 것이다. 그리고 다른 사람들에 의해 설계된 에이전트들과 통신할 경우에는 같은 사투리(dialect)를 사용하고 있는가를 확인해야 한다.

◎ 이달의 추천 시스템

KQML로 대화하는 Java 애플릿을 구성할 수 있는 셸(shell)이 있다. Java Active Template(JAT)*라 하는데, 다운로드하여 사용해볼길 바란다.

1. R.S. Patil, et al., "The DARPA Knowledge Sharing Effort: Progress Report," *Proc.Third Intl Conf. on Principles of Knowledge Representation and Reasoning*.
2. J. Mayfield, Y. Labrou, and T. Finin, "Evaluation of KQML as an Agent Communication Language," in *Intelligent Agents II: Agent Theories, Architectures, and Languages*, J.P. Muller and M. Tambe, eds, Springer-Verlag, Berlin, 1996.

*URLs

KQML : www.cs.umbc.edu/kqml/kqmlspec/spec.html

FIPA : drogo.cselt.stet.it/fipa/

KIF : hpdce.stanford.edu/newkif.html

JAT : cdr.stanford.edu/ABE/JavaAgent.html

«IEEE Internet Computing, Vol. 1, No. 2,
March April 1997»

.....

본 기사는 인하대학교의 유상봉 편집위원이 "IEEE Internet Computing"에서 발췌하였으며 출판사인 IEEE Computer Society의 연락처는 다음과 같다.

• <http://computer.org/internet/>