論文
3-4

계산논리기반에이전트를위한에이전트통신언어에관한연구
AStudyonAgentCommunicationLanguagesfor ComputationalLogic-basedAgents

이 명 진(Myung-Jin Lee)[1] 한현관(Hyun-Goun Han)[2]

요약

FIPA 커뮤니티는 에이전트와 관련한 표준들을 개발하고 있는데, 이들 중 가장 중요한 것은 에이전트 통신 언어(Agent Communication Language: ACL)이다. 이 언어는 명확하게 협상을 지원하도록 의도된 많은 통신 행위(communicative act)들을 포함하고 있다. 이 논문에서는 자신의 목표를 달성하기위해 자원들을 교환하는 멀티-에이전트 환경을고려하고, 에이전트를 계산 논리기반 형식화로 표현하고, 그리고 이러한 계산 논리기반 에이전트가 사용하는 협상을 위한 ACL을 간단히 서술한다. 특히, 여기서는 상대방 에이전트가 요청(제안)의 이유 혹은 거절(거부)의 이유를 알 수 있게 하는 몇몇 통신 행위들을 소개하고 이들을 평가한다.

Abstract

The FIPA community is developing a range of agent-related standards, of which the centerpiece is an Agent Communication Language (ACL). This language includes a number of communicative acts explicitly intended to support negotiation. In this paper, we consider a multi-agent environment that exchanges the resources for achieving agents' goals, represent agents as computational logic-based formalizations, and describe a simple ACL for negotiation using logic-based agents. In particular, we introduce and compare some communication acts that enable counter-agents to know the reason of request(proposal) or the reason of rejection(refusal).

Key words: 에이전트(Agent), 에이전트 통신 언어(Agent Communication Language), 계산 논리(Computational Logic), 협상(Negotiation).

1) 정회원:아시아대학교 IT마스터학과
2) 정회원:대구산업정보대학 컴퓨터정보계열

논문접수 : 2004. 3. 5.
심사완료 : 2004. 3. 19.

1.서론

자율적인 에이전트와 멀티 에이전트 시스템(Multi-Agent Systems: MAS)은 공동체 혹은 조직의 일부로써 상호 작용할 수 있는 자율적인 개체들을 모델링하는 메타포어로 채택되어 왔다. 최근에 계산 논리기반 인공지능은 MAS 개발로의의미있는 연구들을 만들기 시작하였는데, 실제로 계산 논리기반 형식화들은 에이전트의 지식과 이러한 지식을 사용하고 갱신하는 에이전트의 추론을 모델링하고 구현하는 강력한 방법이다. 계산 논리는 에이전트와 에이전트들의 공동체를 모델링할 수 있게 하고, 시스템 구현의 기초로 사용될 수 있는 연산적인(operational) 의미론을 제공한다[12].

Sadri, Toni, 그리고 Torroni는 자원 협상을 위한 에이전트 대화로의 계산 논리기반 접근방법을 소개하였는데, 논리기반 환경에서 에이전트는 목표, 의도, 그리고 믿음을 포함하는 논리기반 지식 표현을 가진다. 에이전트는 자신의 목표 달성에 어떤 자원이 부족할 경우에, 대화 프리미티브(*dialogue primitive*)들을 사용하여 대화를 수행하고 부족한 자원을 요청하여 대화기반 협상 프레임워크에서 다른 에이전트들과 협상한다. 또한 에이전트는 계산 논리기반 증명 절차에 기초하여 어떤 프리미티브를 전달할 것인가를 결정한다[11].

한편, 에이전트통신 언어(ACL)의 주된목적은 이질적인 에이전트들이 상호 작용할 수 있게 하는 적당한 프레임워크를 모델링하고, 환경과 지식에 관한 정보를 전달하는 의미있는 문장들을 교환하는 것이다[1]. 따라서 ACL은 다음과 같은 바람직한 여러 가지 특징들을 가져야 한다[2]: 에이전트는 자신의 환경과 목표에 대한 지식을 관리하고 사용하기 때문에 ACL은 지식 표현 언어이어야 하며, 선언적이고 구문적으로 간단하여야 하며, 통신 행위에 대한 정확한 의미론(semantics)을 제공하여야 하며, 그리고 네트워크 환경에 적합한 언어이어야 한다.

이 논문에서는 환경의 상태에 대한 에이전트 의정신적 태도(mental attitude)들에 초점을 두고, 자신의 목표를 달성하기 위해 자원들을 교환하는 멀티-에이전트 환경을 고려한다. 에이전트의 지식을 선언적인 계산 논리기반 형식화로 표현하고, 자원이 한정된 계산 논리기반 에이전트가 사용하는 협상을 위한 ACL을 간단히 서술한다. 특히, 여기서는 상대방 에이전트가 요청(제안)의 이유 혹은 거절(거부)의 이유를 알 수 있게 하는 몇몇 통신 행위들을 소개하고 이들을평가한다.

2.협상언어

협상의 범주를 협상 언어, 협상 결정, 그리고 협상 과정으로 분류할 수 있는데, 여기서는 협상 언어에 관심을 가진다. 협상 언어는 협상을 위한 통신 프리미티브들, 이들의 의미론, 그리고 협상 프로토콜에따른 이들의 사용법에 관한 연구 분야이다. 이러한 협상 언어는 전송자의 의도를 수신자에게 전달하는 담화 행위 이론에 기초를 두고 있다.

통신 프리미티브들을 각각 에이전트들의 협상 시작, 에이전트들의 주어진 문장에 대한 반응, 그리고 에이전트들의 협상 종료에 대응하는 세 가지의 부류들로구별할 수 있다 *initiators, reactors, completers.* 상호 작용은 *initiator* 프리미티브로 시작하며, 상호 작용 동안에 *reactor* 프리미티브가 사용되고, 마지막으로 *completer* 프리미티브에 의해 상호 작용이 종료한다[8].

FIPA ACL은 잘 정의된 형식적인 의미론적 기초를 가지며 메시지들은 일련의 메시지 매개변수들을 가지는데, 효율적인 에이전트 통신에 필요한 이들 매개변수들은 환경에 따라 변경될 수 있다. 다음은 FIPA ACL 메시지의 일반적인 요소들을 나타낸다:

```
(performative
        :sender
        :receiver
        :content
        :language
        )
```

또한 FIPA ACL 통신 행위들을 다음과 같이

분류할 수 있다[3]: *action performing, error handling, information passing, negotiation, requesting information.* 이 논문에서는 이들 중 협상에 사용될 수 있는 *action performing* 부분에 해당하는 *request* 통신 행위와 *negotiation* 부분에 해당하는 *refuse* 통신 행위에 초점을 둔다.

KQML은 지능적인 에이전트들 사이의 고수준의 통신을 지원하기 위한 언어와 이와 관련된 프로토콜들이다. KQML을 세 가지의 계층들(*content, message, communication* 계층)로 나눌 수 있는데, 이들 중 *content* 계층은 전달될 지식을 어떤 언어로 표현한 식을 포함한다[6]. KQML 메시지의 일반적인 요소들은 FIPA ACL의 그것과 매우 유사하다. 여기서는 협상에 사용될 수 있는 *effector* performative *achieve*와 *informative* performative *tell*에 초점을 둔다.

한편, Sadri, Toni, 그리고 Torroni는 자원획득을 위한 두 개의 에이전트들 사이의 협상 대화를 생성하는데 필요한 언어, 지식, 그리고 추론을 서술하였다. 그들은 가설(abducible, hypothesis)로 취급되는 대화 performative와 더불어, ALP(Abductive Logic Programming)처럼 논리 프로그램과 무결성 제약조건으로 에이전트의 믿음을 표현하였다[11]. 대화 프리미티브는 *tell*(*a, b, Move, t*) 형태이며, 여기서 *a*와 *b*는 각각 전송 에이전트와 수신 에이전트, *t*는 프리미티브가 발생한 시간, 그리고 *Move*는 아래와 같이 순환적으로 정의된 대화 내용(move)이다*request*(*give*(*R*)), *promise*(*give*(*R*), *give*(*R*)), *accept*(*Move*), *refuse*(*Move*), *challenge*(*Move*), *justify*(*Move, Support*).

마지막으로, Wooldridge와 Parsons는 멀티-에이전트 협상을 위한 논리기반 언어들의 사용을 고려하고, 두 가지의 계산상의 문제들을 정의하였다*success problem*과 *guaranteed success problem*. 또한 그들은 위의 문제들에 기초하여 협상을 위한 세 가지의 언어들을 서술하였다: *classical propositional logic, language for electronic commerce, negotiation meta-language*[13]. 이 논문에서는 FIPA ACL이 제공하는 협상 프리미티브들에 기초한 협상 메타-언어를 고려하고, 협상에 사용되는 illocution들(*request, reject*)에 초점을 둔다.

다음 절에서는 협상 과정동안에 상대방 에이전트가 요청(제안)의 이유 혹은 거절(거부)의 이유를 알 수 있게 하는 여러 가지 통신 행위들을 비교한다: FIPA ACL 통신 행위들(*request, refuse*), KQML performative들(*achieve, tell*), 대화 내용들(*request, refuse, challenge, justify*), 협상 illocution들(*request, reject*), 자동 추론 시스템에 사용되는 argumentation, 그리고 여기서 소개하는 메타-술어들(*request, reject*).

3. 협상을 위한 통신 행위

메시지를 수신한 상대방 에이전트가 요청(제안)의 이유 혹은 거절(거부)의 이유를 알 수 있다면, 이러한 지식은 아래와 같은 여러 가지 목적들로 활용될 수 있을 것이다:
상대방 에이전트를 신뢰할 경우에 에이전트들과의 통신은 세상에 관한 믿음들 혹은 다른 에이전트들에 관한 믿음들을 만들 수 있다.

세상, 다른 에이전트들, 그리고 자기 자신에 관하여 에이전트가 갖고있는 지식으로부터 연역된 다른 믿음들을 만들 수 있다.

요청의 이유 혹은 거절의 이유를 통해 상대방 에이전트의 행위 혹은 목표에 대해 가설적으로추론할 수 있다.

FIPA ACL *request* 통신 행위는 수신 에이전트가 어떤 행위를 수행하기를 요청한다. 즉, 수신 에이전트가어떤 행위를 수행한 다음 그에 따른 다른 어떤 통신 행위를 보내줄 것을 요청함을 의미한다. 예를 들어, 에이전트 *i*가 에이전트 *j*에게 못을 달라고 요청한다면 이것을 다음과 같이 간단히 서술할 수 있다:

```
(request
        :sender   i
        :receiver j
        :content
         "give(nail)"
        :language prolog)
```

수신 에이전트가 요청한 행위를 거절하는경우

에는 요청한 행위와 더불어 거절의 이유에 대한 설명을 *refuse* 통신 행위를 통해 전달한다. 예를 들어, *j*가 *i*로부터 못을 달라는 요청을 받았지만 자신의 목표를 달성하기 위해 못을 건네줄 수 없다면 이것을 다음과 같이 간단히 서술할 수 있다:

```
(refuse
        :sender   j
        :receiver  i
        :content
         "give(nail)"
         "have_to(nail)"
        :language prolog)
```

만약 *i*가 *request* 통신 행위를 전송할 때 못이 필요한 이유를 함께 *j*에게 전달한다면, *j*는 거절의 이유와 더불어 *i*의 못을 이용한 문제 해결에 대한 다른 해결책을 제공할 수도 있을 것이다.

KQML 에이전트는 믿음들과 목표들로 구별되는 자신의 가상 지식베이스(Virtual Knowledge Base: VKB)를 관리하며, *achieve* performative는 수신 에이전트가 :*content* 매개변수에 있는 문장을 참으로 만들도록요청한다. 수신 에이전트는 자신의 VKB로부터 :*content* 매개변수에 있는 문장이 유도되는가를 검사하고, 그 결과를 *tell* performative의 :*content* 매개변수에 문장의 형태로 서술하여 전송한다. 해당 문장을 참으로 만든다면, 이것은 단순히 그 문장이 에이전트의 VKB에 있음을 의미한다. 이와 같이 *achieve* performative에서는 문장을 참으로 만들어야 하는 이유에 대해서는 아무런 언급을 하지 않는다.

Sadri, Toni, 그리고 Torroni에서 에이전트들이 발생시킨 대화 내용들은 순서대로 공유된 블랙보드에 기록된다(대화 저장). 예를 들어 다음과 같은 대화 저장을 고려해보자:

tell(*i*, *j*, *request*(*give*(*nail*)), 1)

tell(*j*, *i*, *refuse*(*request*(*give*(*nail*))), 2)

tell(*i*, *j*, *challenge*(*refuse*(*request*(*give*(*nail*)))), 3)

tell(*j*, *i*, *justify*(*refuse*(*request*(*give*(*nail*))), {~*have*(*nail*)}), 4)

여기서 술어의 마지막 인수는 트랜젝션 타임스탬프이고, 대화가 진행됨에 따라 새로운 대화 내용들이 발생하기 때문에 대화 저장은 단조적으로 증가하게 된다. 위의 대화 저장에서 세번째 대화는 요청 '*request*(*give*(*nail*))'의 거절 이유를 물어보며, 네번째 대화는 요청한 자원을 가지고 있지 않기 때문에 그 요청을 거절함을 알려준다(~*have*(*nail*)). 이와 같이 여기서는 요청의 이유에 대해서는 언급이 없으며, 요청을 하고 그것의 거절 이유를 알기까지는 세 개의 추가적인 트랜젝션들이 요구된다.

한편, 자동 추론 시스템에 해당하는 argumentation 시스템을 고려해보자[9, 10]. 고전적인 논리에서 argument들은 결론에 이르게 하는 추론들의 나열이며, argument가 옳다면 그 결론은 참이다. Argumentation 형식을 다음과 같이 간단히 표현할 수 있다: (*Sentence*, *Grounds*). 여기서 *Grounds*는 *Sentence*를 추론하는데 사용된 사실들과 규칙들을 나타내는 레이블들의 집합일 수 있다. Parsons와 Jennings의 연구에서 에이전트 *i*는 자신의 지식베이스의 어떤 부분집합 *i*를 사용하여 자신의 의도들 중의 하나인 *Iti*(*a*)에 찬성하는 argument (*Iti*(*a*), *i*)를 생성하여 이것을 다른 에이전트 *j*에게 전송한다. 여기서 *i*에 포함된 내용은 *i*의 지식베이스에 포함된 사실들과 규칙들을 나타내는 레이블들이다. 이제 *j*는 이들 사실들과 규칙들을 자신의 지식베이스에 추가한 다음에, *i*가 의도하고 제안한 행위 *a*에 대해 찬성하는 argument와 반대하는 argument를 생성하고 rebutting과 undercutting의 개념을 사용하여 그 제안을 평가한다. 따라서 argumentation 시스템은 임시적인 증명의 개념을 이용하며, 에이전트들은 협상을 통해 상대방 에이전트의 제안에 찬성하는 사실들과 규칙들을 알 수 있게 된다.

협상 메타-언어의 협상 illocution들 *request*(*i*, *j*,)와 *reject*(*i*, *j*,)에서 는 전자상거래를 위한 언어 *L*1의 식이다. 형식적으로 *L*1은 협상 이슈들의 유한한 집합을 포함하는 일차 논리의 부분집합이다. 예를 들어, 중고차 구매에 대해 협상하는 두 개의 에이전트들 사이의

협상 히스토리를 가정해보자:

request(*i*, *j*, (*price* < 4000) (*model* = ?) (*age* = ?))

offer(*j*, *i*, (*price* = 3500) (*model* = *Escort*) (*age* = 8))

reject(*i*, *j*, (*price* = 3500) (*model* = *Escort*) (*age* = 8))

offer(*j*, *i*, (*price* = 3900) (*model* = *Golf*) (*age* = 6))

위의 협상 히스토리에서 단지 에이전트들은 변수 형태인 협상 이슈들(price, model, age)에 대해 서로가 만족하는 고정된 값(정수)을 만들려고 노력한다.

마지막으로, 에이전트들이 부족한 자원의 할당에 관해 협상한다고 가정한다면 에이전트는 자신의 목표 달성에 필요한 부족한 자원의 할당을 요구한다. 이러한 경우에 자원이 한정된 에이전트를 위한통신 언어를 간단히 아래와 같이 메타-언어 형식으로 표현할 수 있다[7]:

request(*i*, *j*, *g*, *r*)

reject(*i*, *j*, *g*, *r*)

위에서 *request*(*i*, *j*, *g*, *r*)은 *i*가 자신의 목표 *g*를 달성하는데 필요한 부족한 자원 *r*을 *j*에게 요청하는 통신 행위인데, *j*는 왜 *i*가 자원 *r*을 필요로 하는가를 *g*를 통해 알 수 있다. 한편, *reject*(*j*, *i*, *g*, *r*)은 이전에 *j*가 *i*로부터 수신한 *request*(*i*, *j*, *g*, *r*)을 해석한 결과로 생기는 통신 행위들 중의 하나인데, *j*는 자신의 목표 *g*을 달성하기 위해 *r*이 필요하기 때문에 *i*에게 *r*을 넘겨줄 수 없음을 알리는 통신 행위이다. *request*와 *reject* 통신 행위를 통해 수신 에이전트는 상대방 에이전트의 목표 혹은 부목표(subgoal)를 알 수 있기 때문에, 자신과 상대방 에이전트에 관한 추론 능력을 향상시킬 수 있으며, 요청을 거절하는 경우에는 명확한 이유뿐만 아니라 요청을 해결할 수 있는 다른 해결책까지도 알려줄 수 있다.

4.평가및연구과제

이 논문에서는 에이전트를 계산논리기반 형식화로 표현하고, 이러한 계산 논리기반 에이전트가 사용하는 협상을 위한 ACL을 간단히 서술하였다. 특히, 상대방 에이전트가 요청(제안)의 이유 혹은 거절(거부)의 이유를 알 수 있게 하는 몇몇 통신 행위들을 살펴보았다.

FIPA ACL과 KQML은 각각 *request*와 *achieve* 통신 행위를 제공하지만, 이 통신 행위를 수신한 FIPA 에이전트와 KQML 에이전트는 왜 이러한 요청을 송신 에이전트가 했는가에 대해서는 모른다. 특히, KQML 에이전트는 단지 자신의 지식베이스로부터 전송된 문장이 유도되는가를 검사한다. FIPA ACL *refuse*와 *reject-proposal* 통신 행위들은 제안한 요청에 거절하는 이유를 *':content'* 부분에 기술할 수 있다. 한편, 공유된 블랙보드 대화 저장에서 요청이 거절되었을 때, 상대방 에이전트가 그 이유를 아는데는 추가적인 트랜젝션들이 필요하며, 일대일 협상 에이전트들 사이의 대화를 생성하기 위해 abductive 증명-절차의 동적인 수행을 사용한다. Parsons와 Jennings의 접근방법은 일대일 협상을 위한 유연한 프로토콜들을 정의하는데, 그들의 argumentation 시스템은 임시적인 증명의개념을 사용하여 상대방 에이전트의 제안에 찬성하는 사실들과 규칙들을 알 수 있게 한다. 하지만 문장을 추론하는데 사용된 사실들과 규칙들의 집합이 매우 클 경우에, 일반적으로 다음 번의 argument의 크기가 점진적으로 증가하는 경향을 보이기 때문에 argumentation 시스템은 통신상의 심각한문제를 야기시킬 가능성이 높다.

모든 환경에 적합한 통신 언어와 협상 프레임워크를 고안하는 것은 불가능할 수도 있다. 3절의 협상 메타-언어에서 에이전트들은 단지 변수 형태인 협상 이슈들에 대해 서로가 만족하는 고정된 값을 만들려고 노력하였다. 따라서 이러한 형태의 언어는 전자상거래 혹은 경매 등에 적합할 것이다. 만약 에이전트들이 부족한 자원의 할당에 관해 협상한다면, 통신 행위를 통해 상대방 에이전트의 추론 능력을 향상시킨다는 측면에서 이 논문에서 간단히 소개한 메타-언어 형식이 적합할 것이다. 요청의 이유 혹은 거절의 이유는 Kowalski와 Sadri의 *'observe-think-act* 에이전트 사이클'[5]에서 *'observe'*에 해당될 수 있기 때문에, 이것은 다른 에이전트에대한 가설적인 추론을 수행할 수

있는 '가설적 논리 프로그래밍(Abductive Logic Programming: ALP)'[4]에 이용될 수 있을 것이다.

마지막으로, 향후 연구과제들은 다음과 같다: (1) 협상 제안과 협상 규칙을 위한 언어의 필요성협상 에이전트들이 규칙들에 관해 추론할 수 있도록 규칙들을 표현하기 위한 선언적인 언어가 필요한데, 에이전트들은 같은 프로토콜을 구현하여 협상에참여하지만 그들은 규칙들에 관한 추론에 기초하여 그들의 전략들을 조절할 수 있을 것이다. (2) 확장된 협상 프레임워크의 필요성협상에 관련된 다수 에이전트들의 일치들을 처리할 수 있도록 프레임워크를 확장시키는 연구가 필요할 것이다. (3) 자동 협상에 관한 연구의 필요성일대일 협상에서부터 경매와 이중 경매까지의 폭 넓고 다양한 시장 미케니즘들에 관한 보다 일반적인 접근방법이 필요하다.

참고문헌

[1] Chaib-draa, B. and Dignum, F. Trends in Agent Communication Language. *Computational Intelligence*, 18(2). 2002.

[2] Finin, T., Labrou, Y., and Mayfield, J. Desiderata for Agent Communication Languages. In *Proceedings of the AAAI Symposium on Information Gathering from Heterogeneous Distributed Environments*.1995.

[3] Foundation for Intelligent Physical Agents (FIPA). FIPA Communicative Act Library Specification. *http://www.fipa.org/specs/fipa00037/SC00037J.html*. 2002.

[4] Kakas, A. C., Kowalski, R. A., and Toni, F. The Role of Abduction in Logic Programming. *Handbook of Logic in AI and Logic Programming*, 5:235-324. 1998.

[5] Kowalski, R. A. and Sadri, F. From Logic Programming to Multi-Agent Systems, *Annals of Mathematics and Artificial Intelligence*, 25:391-419. 1999.

[6] Labrou, Y. and Finin, T. A Proposal for a New KQML Specification. *Technical Report CS-97-03*, Computer Science and Electrical Engineering Department, University of Maryland Baltimore County. 1997.

[7] Lee, M. J. and Kim, J. S. A Logic Programming Framework for Negotiation among Resource-bounded BDI Agents. In *Proceedings of the International Conference on Intelligent Agents, Web Technologies, and Internet Commerce*. 2001.

[8] Muller, H. J. Negotiation Principles. *Foundations of Distributed Artificial Intelligence*, pp. 211-229. 1996.

[9] Parsons, S. and Jennings, N. R. Negotiation through ArgumentationA Preliminary Report. In *Proceedings of the International Conference on Multi-Agent Systems*. 1996.

[10] Parsons, S., Sierra, C., and Jennings, N. R. Agents that Reason and Negotiate by Arguing. *Journal of Logic and Computation*, 8(3):261-292. 1998.

[11] Sadri, F., Toni, F., and Torroni, P. Logic Agents, Dialogues and Negotiation: An Abductive Approach. In *Proceedings of the Symposium on Information Agents for E-Commerce*. 2001.

[12] Torroni, P. and Toni, F. Extending a Logic Based One-to-One Negotiation Framework to One-to-Many Negotiation. In *Proceedings of the Second International Workshop on Engineering Societies in the Agents World*. 2001.

[13] Wooldridge, M. and Parsons, S. Languages for Negotiation. In *Proceedings of the Fourteenth*

European Conference on Artificial Intelligence. 2000.

한현관

2002년 대영정보(대표)
2002년 대구대학교 대학원 산업정보학과(공학석사)
2003년 (현재) 영남대학교 대학원 컴퓨터공학과(박사과정)
2003년 (현재) 대구산업정보대학 겸임교수
관심분야: 소프트웨어공학, 에이전트, 웹서비스
E-mail: hanhyoun@kms.tpic.ac.kr

이명진
1994년 계명대학교 대학원 컴퓨터공학과(공학석사)
2001년 계명대학교 대학원 컴퓨터공학과(공학박사)
2003년 (현재) 아시아대학교 인터넷비즈니스학과 전임교수
관심분야: 에이전트 시스템, 전자상거래
E-mail: mjleekor@korea.com