

계층적 작업흐름에 의한 멀티 에이전트 조정

박정훈^o, 안세용, 최중민
한양대학교 전자계산학과

Multi-Agent Coordination by Hierarchical Workflow

Jung-Hoon Park, Se-Yong An, Joongmin Choi
Dept. of Computer Science and Engineering, Hanyang University

요 약

최근의 에이전트 시스템 연구는 간단한 작업을 처리하는 단일 에이전트들이 모여 하나의 서비스를 제공하는 멀티 에이전트 시스템 분야에서 활발히 이루어지고 있다. 이러한 멀티 에이전트 시스템에서는 요구된 서비스를 구성인들이 효율적으로 협동하여 제공하기 위해 이들을 통합하고 실행순서 등을 조정하는 것이 중요하다.

본 논문에서는 계획, 그룹, 작업의 세가지 단계로 구분되는 규칙과 ECA 규칙을 사용하여 계층적 작업흐름을 정의하고 이를 통하여 에이전트들을 조정하는 방법을 제안한다. 계층적 작업흐름의 사용으로 기존의 방법과 달리 에이전트들 계층화하여 새로운 서비스를 구성할 경우 세부적인 수행사항에 대한 언급없이 상위 단계의 계획 규칙만을 지정하여 구성할 수 있는 장점이 있다.

1. 서 론

에이전트란 지능을 갖고 필요에 따라 다른 에이전트와 통신을 통하여 사용자의 일을 대신해 주는 자율적인 프로그래머라 할 수 있으며 일반적으로 하나의 독립된 에이전트는 간단한 작업을 처리한다. 이러한 에이전트들은 보다 큰 작업을 처리하기 위해 그룹화되어 멀티 에이전트(multi agent) 시스템을 구성하게 된다. 이와 같은 멀티 에이전트 시스템에서는 요구된 작업을 에이전트들이 효율적으로 협동하여 하나의 공동된 목적을 달성하기 위해 여러 에이전트들을 통합하고 이들의 작업 순서 등을 조정하는 것이 필요하다. 이것이 에이전트 조정(coordination)이다.

현재 멀티 에이전트 시스템에서의 조정에 관하여 많은 방법론이 제안되었다. 대표적인 것으로는 블랙보드를 이용하는 방법, 계약(contract)에 의한 방법, 멀티 에이전트 계획(multi agent planning) 및 협상(negotiation)에 의한 방법 등이 있다[1].

본 논문에서는 워크플로우 도메인에서의 멀티 에이전트 조정을 위하여 계층적 계획 규칙(hierarchical plan rule)과 ECA(Event/Condition/Action) 규칙을 이용하는 방법을 제안한다. 이 방법은 에이전트들이 하나의 서비스를 제공하기 위하여 계획 규칙 및 역할(role) 정보를 이용, 계층적 작업 흐름을 정의하고 이를 바탕으로 개층적 시스템 구조를 구성하여 서비스를 제공하게 된다.

2. 관련 연구

TrnGS_{DM}[2, 3]는 오스트리아의 Linz 대학에서 연구된 워크플로우 시스템으로 조직체의 민원민 회화 요구에 대처하기 위해 유연하고 동적인 반응을 허용하는 개념들을 제공하고 있다. 이를 위해 TrnGS_{DM}는 객체지향 데이터베이스 기술, 역할 및 ECA 규칙 등의 기술이 통합되었다. 이 시스템은 ECA 규칙에 의한 조정 방법을 깨울

회 함으로써 조직체의 지식 특이 특정 비즈니스 프로세스에 독립적으로 표현될 수 있게 하였다. 이것은 조정 방법의 변경과 이의 제사용성을 용이하게 해주는데 조정 방법은 활동 집렬, 에이전트 선택 및 워크리스트 관리의 세 영역에 포함되어 있다.

SAMOS[4]는 스위스의 Zurich 대학에서 심입용 COOBMS가 ECA 규칙 형태에 의해 명서되는 반응적인 행동을 갖도록 확장한 연구 프로토타입이다. SAMOS는 이벤트가 발생하면 해당 이벤트에 대응하는 규칙이 존재하는지를 규칙베이스에서 검사하여 존재할 경우 규칙을 실행하게 된다. SAMOS기반 협동 에이전트 환경에서는 브로커들이 이를 기반으로 조정 및 통신을 수행하여 작업을 처리하게 된다.

3. 기존의 문제점 및 해결 방안

앞서의 TrnGS_{DM}나 SAMOS기반의 방식은 어느 특정 서비스와 관련된 멀티 에이전트 시스템을 구축할 때 긴체적인 흐름의 세부적인 상황까지 직성해야만 한다. 이는 모든 에이전트들이 단일 계층을 이루며 ECA 규칙만을 사용하고 있기 때문이다.

본 연구에서의 에이전트 조정은 TrnGS_{DM}와 SAMOS기반의 방식과 마찬가지로 ECA 규칙을 기반으로 하고 있다. 그러나 ECA 규칙뿐만 아니라 계획 규칙도 사용하며 시스템을 구성하는 에이전트들을 그들의 능력에 따라 등급을 달리하며 계층적으로 그룹화하였다[그림 1]. 에이전트 상호관계에 따른 미리 정의된 계획 규칙의 사용은 에이전트간 통신을 최소화하며 충돌없는 실시간 작업 수행을 보장하고 계층적 그룹화는 이전의 방식과 달리 새로운 서비스를 구성할 경우 세부적인 수행사항에 대한 언급없이 상위 단계의 계획 규칙만을 지정하여 구성할 수 있게 한다. 워크플로우의 예에서 보면 사용자나 개발자는 전체 서비스 뒤싱에 필요한 큰 단위의 작업 순서나 계한 사항만을 기술하도록 하고, 각 단위의 작업을 이루기 위한 서

부 사항과 절치, 그리고 세부 절차에 요구되는 제한 사항 등에 관한 규칙은 그것을 진담하는 에이전트 그룹을 생성하여 그 안에서 해결하게 된다

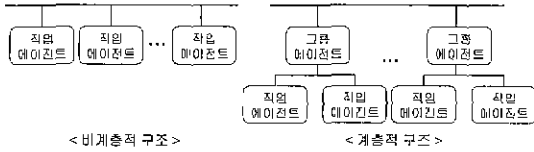


그림 1 비계층적/계층적 에이전트 구조

층적인 구조를 이루며 각기 별도의 흐름을 제어하게 된다 또한 작업 에이전트의 역할을 나타내는 역할 정보도 정의되었다

각 규칙과 역할 정보는 다음과 같다

- ① 계획 규칙(Plan Rule) - 워크플로우의 기본적인 작업 순서를 정의하며 SH는 이 규칙에 따라 기본적인 작업 그룹을 실행한다

```

DEFINE PLAN_RULE           예) 국외출장 신청
RULENAME <rule_name>     DEFINE PLAN_RULE
PERFORM                   RULENAME business_trip_abroad
LISTRULE                  PERFORM
                           LISTRULE
                           business_trip_doc
                           business_trip_info
    
```

- ② 그룹 규칙(Group Rule) - 에이전트 그룹 단위의 작업 순서를 정의한다 그룹 규칙은 에이전트들이 처리할 작업들을 그룹 내 에이전트의 공동작업으로 해결하기 위하여 제어자를 포함한 작업 규칙들로 자세한 순서를 정의한다

```

DEFINE GROUP_RULE
RULENAME <rule_name>
PERFORM
LISTRULE
<task_rule_list> | <controller>
    
```

```

예) 국외출장 관련 문서 작성
DEFINE GROUP_RULE
RULENAME business_trip_doc
PERFORM
LISTRULE
business_trip_application_doc // 출장신청서 작성 작업
detail_schedule_doc // 세부일정계획서 작성 작업
QUERY select_doc business_trip_application_doc ? //일정정보를 얻음
conference_doc technical_doc // 일정정보로 누 작업중 신비
insurance_doc // 보험문서 작성 작업
    
```

- ③ 역할 정보(Role Data) - 그룹 내의 에이전트들이 처리할 수 있는 능력 리스트로 그룹 에이전트가 생성되면서 DF에 등록한다

```

DEFINE GROUP_ROLE
ROLE_NAME <role_name>
LIST <role_list>
    
```

- ④ 작업 규칙(Task Rule) - 작업 에이전트들의 작업 처리방식을 정의하는 규칙으로 일반적인 리스트 형태의 규칙과 ECA 규칙의 기본 형식을 따른다

```

DEFINE TASK_RULE
RULENAME <rule_name>
PERFORM
LISTRULE
<lst_rule>
ECARULE
ON <event_message>
IF <condition>
DO <action_rule>
    
```

```

예) 출장신청서 작성
DEFINE TASK_RULE
    
```

4. HyFlow 시스템의 설계

4.1 전체 시스템 구조

HyFlow는 멀티 에이전트 시스템을 기반으로 계층적 규칙을 이용하여 작업을 효율적으로 에이전트들에 분배, 처리하는 워크플로우 시스템이다 HyFlow에서는 시스템 에이전트와 작업 에이전트의 작업 순서를 정의한 규칙 및 작업 역할을 정의한 역할 정보뿐 이용, 작업 흐름을 정의하여 작업을 처리하게 된다.

HyFlow 시스템은 그림 2와 같이 PA(Personal Agent), SH(Service Handler), DF(Directory Facilitator), AMS(Agent Management System) RRA(Rule Retrieval Agent) 등의 시스템 에이전트들과 상위 에이전트로서 에이전트 그룹을 제어하는 그룹 에이전트, 실제 작업을 처리하기 위해 그룹 에이전트 모듈과 작업 규칙이 결합된 작업 에이전트 및 데이터베이스를 관리하며 자료 등을 제공하는 데이터베이스 에이전트로 구성되어 있다[5]

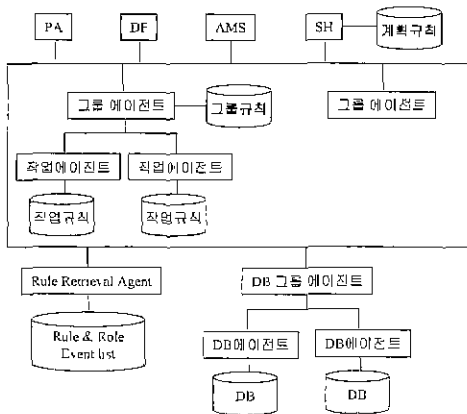


그림 2 계층적 작업규칙을 갖는 워크플로우 에이전트 구조

HyFlow를 이루는 시스템 에이전트들은 서비스의 원활한 제공을 위하여 각각의 특장적인 작업을 수행한다 RRA는 시스템에 필요한 규칙 정보를 검색, 제공하며 이외의 에이전트 기능은 [6]에 언급되어 있다

4.2 HyFlow의 계층적 규칙

4.2.1 계층적 규칙

HyFlow 시스템에서의 계층적 작업흐름은 계획, 그룹, 작업 규칙에 의해 정의되며 이를 바탕으로 서비스를 제공한다 이들 규칙들은 계

```

RULENAME business_trip_application_doc
PERFORM
LISTRULE
  RDQ(DF, User_Info)(Agent_Name) // DF에 user_info를 얻을 agent를
  PERF(AMS, INVOKE, Agent_Name) //요청하고 AMS에 실행을 요청
  GET_INFO(Agent_Name, user_id)(name, pid, depart, )
  REQ(DF, MakeForm)(Agent_Name)
  PERF(AMS INVOKE, Agent_Name)
  MAKE_FORM(Agent_Name, bt_app_doc_web)
  SHOW_FORM(Agent_Name, bt_app_doc_web)
  GET_INFO(Agent_Name, user_id)(country, day, night, ..)
  MAKE_FORM(Agent_Name bt_app_doc)
ECARULE
  ON ASK_USER_INFO(user_id)
  IF EXIST(user_id)
  DO SEND_USER_INFO(user_id)
    
```

다음은 앞서의 규칙을 BNF 형태로 나타낸 것이다.

```

<rule_definition> = DEFINE [ PLAN_RULE | GROUP_RULE
  | TASK_RULE | GROUP_ROLE |
  RULENAME <rule_name>
  [ ( PERFORM <rule_body> ) | ( LIST <role> ) ]
<rule_body> ' = LISTRULE <rule>
  ECARULE <eca_rule>'
<role> ::= <task_rule> { ,<task_rule> } *
<eca_rule> ' = ON <event_message>
  IF <condition>
  DO <action_rule>
<rule> = { { QUERY <capability> <task_rule_name> ?
  <task_rule_name> { <task_rule_name> } * } |
  { CONC <task_rule_name> { . <task_rule_name>
  } + } | <task_rule_name> }
    
```

4.2.2 작업 에이전트

작업 에이전트(Task Agent)는 그림 3과 같은 구조를 가지며 작업을 진행할 경우에는 리스트 형태의 규칙을 참조하고 다른 에이전트로부터 특정 정보 요청이나 작업 요청이 있을 경우에는 ECA 형태의 규칙을 참조하여 실행한다

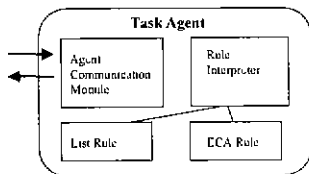


그림 3 작업에이전트의 구조

일반적인 작업 처리의 경우 ECA 형태의 규칙만을 사용하여 수행하면 빈번한 규칙 검색과 조건 전사가 이루어져 작업 속도가 느려지고 효율이 떨어질 수 있다. 본 시스템은 이를 미리 정의된 리스트 형태의 규칙을 사용함으로써 속도 향상 뿐만 아니라 에이전트간 통신을 최소화하며 충분한 실시간으로 작업 수행을 보장하고 있다 ECA 형태의 규칙은 다른 에이전트로부터의 정보 요청 등의 메시지를 받았을때만 사용된다

4.2.3 시나리오

앞의 규칙들을 이용한 작업흐름 제어를 출장신청서 작성부분의 시나리오를 통해 살펴보면 아래와 같다

- 1 사용자가 국외출장신청 서비스를 신청
- 2 PA가 AMS에 SII의 실행을 요청
- 3 SII가 RRA에 국외출장신청 계획규칙을 요청하여 가져옴
- 4 SH가 DF에 BUSINESS_TRIP_DOC을 키리할 그룹에이전트가 어느 것인지 물어옴
- 5 SII가 AMS에 문서 그룹에이전트(Group Agent, GA)에 출장신청 관련 문서작성 작업 실행 요청
- 6 문서 GA가 RRA에 그룹규칙을 요청하여 가져옴
- 7 문서 GA가 AMS에 출장신청서작성 작업에이전트 구분을 요청
- 8 AMS가 에이전트들 구분하고 에이전트가 RRA에서 출장신청서작성 작업규칙을 가져옴
- 9 출장신청서작성 작업에이전트는 사용자정보를 얻을 에이전트들 DF에 묻고 AMS에 실행 요청한 후 사용자 정보를 얻을
10. 출장신청서작성 작업에이전트는 출장비계산은 될 수 있는 에이전트들 DF에 요청 얻을
- 11 출장신청서작성 작업에이전트는 회계 그룹에이전트에 출장비계산을 요청
- 12 회계 GA는 AMS에 출장비계산 에이전트 실행을 요청
- 13 AMS는 에이전트들 구분하고 에이전트는 출장비계산 작업규칙을 읽어옴
- 14 출장비계산 작업에이전트는 출장비계산 정보를 회계 그룹에이전트들 통해 출장신청서 작성 에이전트에 전달
- 15 출장신청서작성 작업에이전트는 출장신청서를 더 만들고 나서 그룹에이전트에 알림
- 16 출장신청서작성 작업에이전트는 다음 작업을 진행

5. 결론 및 향후 연구 계획

기존의 연구에서는 단일 계층의 평면적인 규칙을 이용함으로써 새로운 서비스를 제공할 때마다 이들 각 에이전트들의 작업을 총괄하는데 어려움이 있었다. 본 연구에서는 시스템의 유연성과 확장성을 위해 계층적 작업규칙을 이용한다 계층적 작업규칙을 이용함으로써 새로운 서비스를 제공할 때 하위 에이전트들의 세부적인 규칙들은 미리 정의되어 있으므로 세부적인 수행사항에 대한 내용을 알 필요 없이 상위단계의 계획규칙을 지정할 수 있고, 새로운 작업 에이전트의 추가나 기존의 작업 에이전트의 변경 시에도 전체 시스템의 데이터 흐름에 영향을 주지 않는 유연성을 제공할 수 있게 된다

향후 연구 계획은 규칙베이스의 효율적인 형태를 심의하고 규칙 작성 및 편집 등 관리를 위한 사용자 도구를 개발하고자 한다

[참고문헌]

- [1] H Nwana, L. Lee and N Jennings, "Coordination in Software Agent Systems," BT Laboratories Internal Report, 1994
- [2] G Kappel B Proll, S Rausch-Schott and W Retschitzegger, "TrGS_{new} - Active Object-Oriented Workflow Management," in Proc of the 28th IICSS, pp 727-736, 1995
- [3] G. Kappel, S Rausch-Schott and W Retschitzegger, "Coordination in Workflow Management Systems - A Rule-Based Approach," Coordination Technology for Collaborative Applications - Organizations Processes, and Agents Springer, pp 99-120, 1998
- [4] A Geppert, M Kradolfer and D Tombros, "Realization of Cooperative Agents Using an Active-Oriented Database Management System," in Proc. of the 2nd Workshop on RIDS, Athens, Greece 1995
- [5] FIPA '97 Draft Specification, "Agent Management(TC1)," 1997
- [6] 안세용, 박정훈, 오희국, 김정선, 최승민, "멀티 에이전트 기반 워크플로우 시스템의 설계 및 구현," 한국정보과학회 봄 학술발표 논문집, 제25권, 제1호, pp. 488-490, 1998