

ACPMoel : Adaptive Coordination Policy Model for Cooperative Work on Distributed Environment (ACPMoel : 분산 환경에서 공동 작업을 위한 적응적 조정 정책 모델)

이세훈* 백영태** 서대우***
(Seihoon Lee) (Yeongtae Baek) (Daewoo Seo)

ABSTRACT

Computer Supported Cooperative Work applications such as cooperative distance learning, distance medical, group game, teleconferencing is spreading in all area of real-life. But to develop more various collaborative work environment of private and group, CSCW Applications must have the model that should execute a private and group coordination policy, effectly.

Also, Most of these cooperative work system has a static type that the coordination policy decided the beginning of system development. There is a problem that these static policy model can't change dynamically when perform cooperative work system.

So, In this paper, to solve this problem, we proposed the ACPMoel that can change adaptively a private and group coordination policy in runtime as separate coordination policy and real operations. The proposed model has the cooperate structure of two way bus structure and, make adaptive policy change possible in runtime though designed logic based description language, ACPL, to developers

요 약

CSCW 응용으로는 상호 협력적 원격 교육, 원격 진료, 그룹 게임 분야, 화상 회의 시스템 등으로 실생활의 거의 모든 분야로 확대되고 있다. 그러나 이러한 CSCW 응용이 보다 개인 및 그룹의 다양한 공동 작업 환경을 구현하기 위해서는 개인 및 그룹의 정책을 기술할 수 모델을 갖고 있어야 한다.

그러나, 기존의 연구들은 여러 사람들이 공동 작업을 할 때 각 개인이나 그룹에서의 조정 정책들이 정적으로 개발 초기 단계에서부터 정해져서 개발되어 진다는 문제점을 안고 있다.

따라서 이 논문에서는 이러한 문제점을 해결하기 위한 방법으로는 조정 정책과 실제 연산을 분리함으로써, 개인 및 그룹의 정책을 실행 시간에 적응적으로 변경할 수 있는 모델을 제시했다. 제시한 모델은 이중 버스 구조의 상호 협조 구조를 가지며, ACPL이라는 논리 기반의 명세 언어를 개발자에게 제 공함으로써 실행 시간의 적응적인 정책 변경을 가능하게 한다.

1)

인터넷과 개인용 컴퓨터의 대중화로 통신망 서비스를 이용하는 컴퓨터 사용자들이 많아지고 있다. 이는 지역적으로 분산되어 있는 사람들이 서로의 정보를 공유하여 의사 결정하는 것뿐만 아니라 여러 사용자가 공동 작업을 수행할 수 있

는 형태로 작업 환경이 바뀌고 있다. 이러한 통신 망에 연결된 컴퓨터를 통해 그룹 활동을 지원하고 자 하는 요구로 컴퓨터 지원 협동 작업(CSCW: Computer Supported Cooperative Work) 혹은 그룹웨어(Groupware)라 불리는 새로운 응용 분야가 생겨났다[1,2]. CSCW 응용으로는 화상 회의 시스템, 상호 협력적 원격 교육, 원격 진료, 그룹 게임 분야 등으로 실생활의 거의 모든 분야로 확대되고 있다. 따라서 CSCW 응용을 보다 효율적으로 개발하기 위한 프레임워크에 대한 연구가 활발히 진행되고 있다[3,4,5].

*정회원: 인하공업전문대학 전자계산학과 부교수

**정회원: 김포대학 멀티미디어전공 전임강사

***준회원: (주)케이맥스 기술연구소 실장

그러나 이러한 CSCW 응용이 보다 개인 및 그룹의 다양한 공동 작업 환경을 구현하기 위해서는 개인 및 그룹의 정책을 기술할 수 모델을 갖고 있어야 한다[4,5,6,7].

그러나, 기존의 연구들은 여러 사람들이 공동 작업을 할 때 각 개인이나 그룹에서의 조정 정책(coordination policy)들이 정적으로 개발 초기 단계에서부터 정해져서 개발되어 진다는 문제점을 안고 있다. 이러한 문제점을 해결하기 위한 연구는 현재 많지 않으며, 절차적인 언어를 이용한 방법과 메타레벨의 정의를 이용하는 방법이 있다. 절차적 언어를 이용하는 방법은 GroupKit[B]과 MASH[B]등이 있으며, 정의를 이용하는 방법은 DCWPL[10]이 있다. 그러나 이러한 연구들도 지능적 요소를 포함한 적응적 조정을 하는 데에는 구조적인 한계점을 갖고 있다.

따라서 이 논문에서는 이러한 문제점을 해결하기 위한 방법으로는 조정 정책과 실제 연산을 분리하여 조정 정책을 지식 베이스화 함으로써, 개인 및 그룹의 정책을 실행 시간에 적응적으로 변경할 수 있는 모델을 제안한다. 모델의 중요 개념은 이중 버스 구조의 상호 협조 구조를 가지며, 논리 기반의 명세 언어를 개발자에게 제공함으로써 실행 시간의 적응적인 정책 변경을 가능하게 하는 것이다.

1. 조정 정책에 대한 고찰

이 절에서는 CSCW 응용 환경에서 조정 정책에 대한 개념 및 조정정책을 해결하기 위한 기존의 조정 정책 해결 모델에 대하여 고찰한다.

2.1 조정 정책의 개념

CSCW응용은 많은 사람들 사이에서 일어나는 공동 작업들을 컴퓨터를 이용해서 보다 쉽게 할 수 있게 하는데 목적이 있다[1,2]. 공동 작업은 특정한 정책에 의해서 조정된다. 다시 말해서, 공동 작업에는 달성하려는 목적에 따라서 정책이 달라질 수 있으며 정의된 정책에 따라서 공동 작업에 참여하는 사용자들의 역할과 상호 작용을 제한할 수 있다. 따라서, CSCW가 보다 효과적으로 그룹 공동 작업을 지원하기 위해서는 이러한 규칙들을 소프트웨어 안에 포함시켜야 한다[6,7]. 이러한 규칙들을 조정 정책(coordination policy)이라고 한다. 조정 정책에는 접근 제어(access control), 동시성

제어(concurrency control), 발언권 제어(floor control), 역할과 세션 제한 조건(role/session constraints), 예외 처리(exception handling) 등이 포함된다.

조정 정책은 각각의 그룹들의 일하는 형태나 조직의 구조에 따라서 다를 수 있으며, 같은 그룹이라 하더라도 협력 작업의 단계에 따라서 다른 정책에 따라서 조정될 수 있다. 협력적인 시스템의 개발은 일반적으로 조정 정책의 정의를 위해서 전문가들이나 개발자들 또는 사용자들의 상호 작용을 통해서 이루어진다. 이러한 방법은 모든 정의가 응용 개발 시에 이루어지고 이 정책이 모든 경우에 적용되는 once-and-for-all 정책이라 할 수 있다. 지난 십 수년 동안 CSCW 분야에서는 매우 다양한 공동 작업 시스템과 상업적 제품들을 개발해 왔으며, 이들 대부분이 once-and-for-all 방식을 취하고 있다. 이러한 접근 방식은 응용 개발 시 결정된 정책이 모든 경우에 적용되기 때문에 유연한 공동 작업을 지원하지 못한다. 따라서, 응용과 정책을 분리하는 방식을 도입한 CSCW 응용 개발 방법이 요구된다.

또한, 기존의 응용들은 중앙 집중식 구조는 조정하기는 쉬우나 확장하기 힘들며, 중앙 컴포넌트들은 병목 현상이 생기기 쉬우며, 중앙 서버의 예러는 전체 시스템에 심각한 영향을 미치는 문제점이 있다. 중앙 집중식 구조는 하나의 그룹 정책이 모든 사용자들에게 절대적인 영향을 미친다. 각각의 사용자들은 자신의 정책을 수립하기 힘들며, 따라서 개개인의 정책들이 그룹 활동에 영향을 미칠 수 없다.

2.2 기존의 해결 모델

2.2.1 절차적 언어에 의한 해결방법

전통적인 CSCW 응용의 개발 방식은 C/C++와 같은 절차적 언어를 이용해서 이루어졌다. 따라서, 이러한 응용들은 응용을 개발 할 때마다 매번 반복적이고 똑같은 구현 노력을 들어야만 했다. 그 후, 설계자들은 그룹웨어 시스템에서 필요한 기본적인 서비스들을 추출해서 응용 개발자들에게 GroupKit[B]과 MASH[B]등의 그룹웨어 툴킷(Toolkit)의 형태로 제공하였다. 그룹웨어 툴킷은 다중 사용자 인터페이스를 만들기 위한 프로그래밍 인터페이스의 라이브러리를 제공하고 그룹 통신과 데이터 뷰를 표현과 분리하는 등의 약간의

프로그래밍 추상화를 제공하였다. 이런 방식으로 많은 구현 노력을 줄일 수 있었으나, 조정 정책은 여전히 결차적 언어로 기술되었고 구현 모듈 안에 포함되어서 만들어졌다. 이러한 방식은 여전히 구현 시나 실행 시 구현 정책을 변경할 수 있는 유연성을 제공하지 못했다. 이런 방식으로 개발된 도구들은 일반적으로 다른 도구들과 상호 작용할 수 없다. 이와 같은 이유로 CSCW 응용 사용자들은 그들의 특정 요구를 제대로 만족하는 시스템을 찾기 힘들다.

2.2.2 DCWPL

DCWPL(Describing Collaborative Work Programming Language)은 그룹웨어를 위한 규약 언어이다[2, 10]. DCWPL은 프로그래머가 어떤 전통적인 프로그래밍 언어로 쓰여진 계산적인 프로그램으로부터 분리된 제어 메커니즘을 지정함으로써 조정 프로그램을 정의하는 것을 허용한다. 계산적인 프로그램은 공유 가능한 요소들, 예를 들어 데이터 상태 또는 함수들의 실제적인 설명을 포함하는 반면 조정 프로그램은 사용자가 다른 팀 구성원과 이 요소들을 가지고 상호 작용할 수 있는 방법을 지정한다. DCWPL은 공유 자원들에 대한 공통 태스크의 수행을 조절하기 위한 메커니즘을 제어 및 조정하는 역할을 한다. 이러한 메커니즘의 정의는 여러 가지 요인에 관계가 되는데 예를 들면, 현재 참석자의 그룹, 이들간의 공유되는 자원들, 태스크들, 최종 목표 등이 있다. 이러한 요인들은 적응적으로 변경되는데 수행 시간에 조정이 경신되는 메커니즘이 필요하다.

DCWPL에서는 그룹 내에서 참여자들 간에 발생하는 다양한 상호 작용에 대한 정의가 불가능하며, 각 참여자들의 정책을 정의하지 못하기 때문에 자율적인 그룹 활동에 대한 보장이 되지 못한다.

III. 적응적 조정 정책 모델 설계

이 논문의 모델에서는 조정과 구현을 분리하는 방식의 조정 정책 모델을 제안한다. CAFE에서는 그룹웨어 개발자들에게 두개의 가상적 버스의 개념을 도입한 상호 작용 구조를 제공하며 조정 정책을 모델링하기 위한 ACPL(Adaptive Coordination Policy definition Language)를 정의한다. 또한, ACPL로 정의된 정책을 제어하고 처리하기 위해서 결차적인 방식이 아닌 규칙 기반의

방식을 사용함으로써 보다 적응적으로 변하는 상황에 대해서 유연한 처리가 가능하도록 한다.

3.1 ACPL

ACPL은 그룹 정책과 개인 정책을 정의하기 위한 언어로서, 전체 협력 작업에 대한 정책을 정의하는데 사용된다. CSCW 응용의 개발자는 화이트 보드나 화상 회의와 같은 일반 응용과는 별도로 ACPL로 정책을 정의함으로써 정책이 적응적으로 변경되더라도 일반 응용의 수정 없이 새로운 정책을 도입한 CSCW 응용을 사용자에게 제공할 수 있다. ACPL로 정의된 정책은 크게 그룹 형태에 대한 부분과 그룹에서 사용할 공유 객체의 정의, 그룹에서 필요한 이벤트들의 정의와 각각의 규칙들로 구성된다.

3.2 그룹 형태에 대한 정책 조정자

그룹의 형태는 그룹의 제목, 사용할 역할, 그룹의 단계, 사용할 객체의 리스트, 참여자 수, 그룹의 개방성 여부, 발언권 제어 방법, 사용할 이벤트의 종류들, 그리고 그룹에서의 규칙들로 구성된다.

그룹에서의 역할과 그룹 활동의 단계는 리스트로 정의될 수 있다. 그룹에서 사용되는 객체들과 이벤트들은 객체 정의 리스트와 이벤트 정의 리스트로 구성될 수 있으며, 그룹에 참여할 수 있는 참여자의 수는 전체 참여자 수와 각 역할별 참여자 수로 정의될 수 있다. 그룹 개방성 여부는 PUBLIC과 PROTECTED 중 하나의 속성을 가질 수 있는데 만약 PROTECTED인 경우에는 비밀번호를 이용해서 그룹에 참여하는 사용자들을 제한할 수 있다. 그룹 내 참여자들 간의 발언권 제어는 그룹 의장에 의해서 발언권이 주어지는 TOKEN_CONTROL 방식과 그룹에 참여한 사용자들이 순서대로 발언하는 ROUND_ROBIN 방식, 그리고 자유롭게 자신의 의견을 제시할 수 있는 FREE 방식 중에 하나를 선택할 수 있다.

리스트 1. 그룹 Grammar

```
Group_definition ::= Group identifier :
                    'Title role_list stage_list object_list
                    participantNum group_attribute
                    floor_control event_list rule_definition
role_list ::= ROLES : id_list
stage_list ::= STAGES : id_list
object_list ::= OBJECTS : object_definitionlist | nothing
event_list ::= EVENTS : event_definition | nothing
```

```

id_list ::= identifier, id_list | identifier
object_definitionlist ::= object_definition
                        | object_definition object_definitionlist
participantNum ::= P_N : total_number number_per_roles
number_per_roles ::= (identifier => total_number) +
group_attribute ::= PUBLIC | PROTECTED : password
floor_control ::= TOKEN_CONTROL
                | ROUND_ROBIN | FREE

```

3.3 객체와 이벤트 정의

그룹은 객체를 공유할 수 있으며, 역할에 따라서 공유하는 객체가 다를 수 있고, 같은 객체를 공유하더라도 다른 권한으로 객체를 사용할 수 있다. 객체 정의는 객체 이름과 클래스 이름, 객체를 사용하는 역할과 권한으로 구성된다.

리스트 2. Object Grammar

```

object_definition ::= OBJECT : objectname classname
objectactivity right
objectname ::= NAME : identifier
classname ::= CLASS : identifier
objectactivity ::= USING_BY : identifier
right ::= W | R

```

이벤트는 그룹에 참여하고 있는 사용자들 간의 상호 작용의 과정에서 발생하며 원활한 그룹 활동을 지원하기 위해서는 이벤트의 적절한 처리가 필요하다. 이벤트의 형태는 이벤트의 발생지와 이벤트가 전달될 목적지, 그리고 이벤트를 구성하는 각각의 필드들의 리스트로 구성된다.

리스트 3. Event Grammar

```

event_definition ::= EVENT identifier : simple_event
simple_event ::= event_src event_des event_fieldlist
event_src ::= SRC : context_list
event_des ::= DST : context_list
event_fieldlist ::= FIELDS : fieldlist | nothing
fieldlist ::= FIELD : fieldlist | nothing

```

3.4 그룹 활동에 필요한 규칙 정의

그룹에서 발생하는 사용자들 간의 상호 작용을 적절하고 각각의 참여자들이 전달되는 이벤트에 대해서 어떻게 처리할 것인가를 규칙을 통해서 정의한다. 규칙은 조건과 행위들의 리스트로 구성되며, 조건이 만족하는지를 체크해서 조건이 만족하면 해당 대상자에게 기술된 행위가 이루어진다. 대상은 참여자인 경우에는 자신만으로 국한 지를

수 있으며, 모든 참여자 혹은 특정한 역할을 수행하는 참여자가 될 수 있다. 조건은 특정한 데이터가 입력되거나 출력되는 경우에 해당하며, 이때의 데이터는 세션 행위와 멀티미디어 데이터에 대한 제어 또는 정의된 이벤트 데이터가 될 수 있다.

세션 행위는 참여, 탈퇴 등을 포함하며 현재 그룹의 상태나 멤버의 정보, 발언권 상황에 대한 요청 등이 될 수 있다. 멀티미디어 데이터인 경우에는 해당 커넥터에서 스트림 데이터가 전송되어 오거나 전송하는 경우를 말한다. 각각의 조건이 만족하면 행위 리스트에 있는 행위들이 수행된다.

리스트 4. 그룹에서 사용할 규칙

```

rule_definition ::= RULES : rule_list
rule_list ::= rule_declare | rule_declare rule_list
rule_declare ::= RULE : rule
rule ::= targetlist : condition => actionlist WHEN stage
BY actor
targetlist ::= LOCAL | ALL | rolelist
condition ::= mode :- datatype
mode ::= IN | OUT
datatype ::= session_command | media_data | event_data
session_command ::= JOIN | LEAVE | LIST_MEMBER
                | GET_ACT_INFO | SET_STAGE
                | GET_MEMBER_INFO
                | LIST_POSSIBLE_ROLES
                | IS_MEMBER | TOKEN_OWNER
                | END_ACTIVITY
media_data ::= connector_name
connector_name ::= string
event_data ::= identifier
stage ::= ALL | id_list
actor ::= ALL | id_list
actionlist ::= action_declare | action_declare actionlist
action_declare ::= string

```

3.5 이원화된 조정 정책

공동 작업에 참여하는 사용자들은 모두 공동의 주제를 가지고 그룹에 참여하게 된다. 따라서, 그룹에 대한 조정 정책은 그룹에 참여하는 모든 사용자들의 상호 작용에 영향을 미치게 된다. 또한, 사용자들은 그룹에 참여하는 형태에 있어서 자신의 목적과 정책에 따라서 상호 작용한다. 그러나, 기존의 연구에서는 그룹 조정 정책만 정의하기 때문에 각각의 사용자들의 개별 정책에 대해서는 고려가 되어 있지 않다.

이 논문에서는 그룹 조정 정책과 그룹에 참여하는 각각의 사용자들의 개별 정책을 모두 지원하며, 두 조정 정책의 조율을 통해서 전체 CSCW

응용의 조정 정책이 이루어지도록 한다.

이를 위해서 버스 개념을 도입한다. 버스는 개념적인 영역으로써 그룹 조정 정책이 영향을 미치는 Collaboration 버스와 개인의 정책이 영향을 미치는 Coordination 버스로 구성된다. 그룹에 참여하는 참여자들은 Collaboration 버스에 연결되어 있기 때문에 Collaboration 버스에 부여되는 그룹 조정 정책에 영향을 받는다.

Coordination 버스는 화이트 보드와 비디오 컨퍼런싱과 같은 실제 CSCW 응용과 개인 정책을 제어하기 위한 개인 조정자(Private Coordinator) 사이를 연결하고 있다. 따라서, 각각의 응용들은 Coordination 버스에 연결되어 있기 때문에 Coordination 버스에 부여되는 개인 조정 정책에 영향을 받는다.

3.6 그룹 및 개인 조정 정책

그룹 정책은 그룹 개설자에 의해서 ACPL을 이용해서 정의된다. 그룹 개설자에 의해서 정의되는 그룹 정책은 그룹에 참여하는 모든 사용자들에게 적용되며 그룹 실행의 기본 정책으로 사용된다. 따라서 각각의 개인의 정책에 우선하며, 그룹이 존재하는 한 정책은 유효하며 그룹 조정 정책에서 정의되는 사항은 8가지이다. 즉, 그룹의 이름(Group Name), 그룹에서 사용 가능한 역할(Role), 그룹에서 사용하는 공유 객체, 그룹에 참여하는 전체 참여자 수와 역할별 참여자 수, 그룹의 개방 여부, 그룹의 제목, 발언권 제어 방법, 정의된 이벤트 발생시 처리 방법 등이다.

개인의 정책은 그룹 관리자 또는 다른 사용자에게서 전달되는 이벤트와 커빅터를 통해서 전달되는 데이터에 대한 규칙을 정의한다. 따라서 개인 조정 정책은 규칙 정의가 대부분이다.

IV. 동적 조정 모델의 평가

이 절에서는 제안된 모델을 이용하여 공동 작업의 가상 시나리오를 적용하여 봄으로써, 제안한 모델의 적용 가능성을 보인다.

모델의 적용을 위해, 텍스트에디터를 통해 작성된 ACPL을 CLIPS(C Language Integrated Production System) 문법[11]의 형태로 변환하는 파서를 구현하였고, CLIPS로 된 사실(facts)와 규칙(rules)을 이용하여 전방 추론을 할 수 있는 JESS(Java Expert System Shell) 엔진[12]을 사용

하여, 시나리오에 따라 변환하였다.

4.1 실험 시나리오

실험 환경은 1명의 강사와 2명의 학생 그리고 1명의 청강생으로 이루어진 수업에서 강사가 그룹의 형태와 그룹 정책을 결정하게 한다. 교육이 이루어지기 위해서는 먼저 강사가 세션 관리자를 통해서 그룹을 생성해야 한다. 그룹을 생성하기 위해서는 여러 그룹 정보가 필요하고 그룹 정책이 결정되어야 한다. 그룹 정책이 결정되고 그룹이 생성되면, 학생들과 청강생이 그룹에 참여하고 자신의 정책을 결정한다. 강사는 먼저 Overview of CSCW라는 강의를 개설하기 위해서 세션 관리자를 통해서 그룹을 생성한다. Overview of CSCW 그룹에서는 강사, 학생, 청강생의 3가지 역할을 정의하고, 수업의 단계를 강의, 질문, 토론, 보고 등의 4단계로 정의한다. 그룹에서 사용할 객체는 강사에 의해서 참고가 되는 Textbook1과 강사에 의해서 편집되고 학생에 의해서 열람되는 Textbook2, 그리고 비디오 스트림 데이터의 상호 교환을 위한 스트림 객체와 오디오 스트림 데이터의 상호 교환을 위한 스트림 객체, 그리고 화이트 보드를 위한 스트림 객체로 결정한다. 수업에 참여하는 참여자의 수는 강사 1명, 학생 2명, 청강생 1명으로 제한하고 그룹의 속성은 암호를 통해서 접근할 수 있는 보호 모드로 설정한다. 초기 발언권 제어 형태는 그룹 의장에 의해서 토큰을 이용해서 제어하고, 그룹에서 사용할 이벤트는 수업의 단계의 변경을 위한 이벤트와 토론의 결과를 강사에게 전달하기 위해서 정의된다. 수업의 단계에서 강의 단계인 경우에는 학생들의 발언권 요청을 무시하고 토론 단계에서 학생들이 토론 결과를 전달하겠다고 하는 경우에는 이를 강사가 수신할 수 있게 한다. 또한, 강의 중에는 학생이나 청강생으로부터 영상 데이터는 전달받을 수 있으나 음성 데이터와 화이트보드를 통한 데이터 전달은 원할한 강의를 위해서 규제한다.

4.2 그룹 정책 기술

그룹을 생성하기 위해서 필요한 정보는 그룹 이름, 그룹에서 사용할 역할들의 종류, 그룹의 작업 단계의 종류, 그룹에 참여할 총 참여자의 수와 역할별 수, 그룹의 사용자들에 의해서 사용될 객체들의 정보, 발언권 제어 형태, 그룹의 속성과 발언

권 제어 형태, 그룹에서 특별하게 사용할 이벤트의 정의, 그리고 그룹 활동에서의 규칙들로 구성된다. 이와 같은 그룹 형태와 그룹 정책은 이 논문에서 설계한 ACPL을 이용해서 리스트 5와 같이 기술한다.

리스트 5. 교육 시스템에서의 그룹 정책

```
Classroom : Overview of CSCW
ROLES : Lectuer Student Attendant
STAGES : Teaching Question Discussion Reporting
OBJECTS :
  OBJECT : materia1 Textbook1 USING_BY :
Lectuer R
  OBJECT : materia2 Textbook2 USING_BY :
Lectuer W
  OBJECT : materia3 Textbook2 USING_BY :
Student R
  OBJECT : video Stream USING_BY : ALL W
  OBJECT : sound Stream USING_BY : ALL W
  OBJECT : whiteboard Stream USING_BY : ALL W
P_N : 4
  Lecture => 1
  Student => 2
  Attendant => 1
PROTECTED : $asl123
TOKEN_CONTROL
EVENTS :
  EVENT CommandTeaching
  SRC : Lectuer
  DST : Student Attendant
  FILEDS :
    FIELD : Content
  EVENT ReportResult
  SRC : Student
  DST : Student
  FILEDS :
    FIELD : Content
RULES :
  RULE :
    LOCAL : IN :- REQUEST_TOKEN WHEN
Teaching BY Student
=> Deny Request
  RULE : Lectue : IN :- ReportResult WHEN
Discussion BY Student => Report Result
  RULE : ALL : IN :- CommandTeaching WHEN
ALL BY Lectuer => Comand by Lecture
  RULE : LOCAL : IN :- sound WHEN Teaching
BY Student Attendant =>Prohibit Speech
```

4.3 그룹 생성

리스트 5에서 기술된 ACPL은 파서를 통해서

그룹 생성을 위해 필요한 정보들로 변환된다. 그룹 생성을 위해서 필요한 정보는 세션 관리와 공유 객체 관리를 위한 모듈에 전달될 데이터와 지식 베이스에 저장되어서 그룹 정책 관리에 사용될 규칙들로 구성된다. 이때, 파서는 ACPL 형태를 CLIPS 형태로 변환하기 위해서 리스트 6과 같은 템플릿을 사용한다.

리스트 6. ACPL을 CLIPS로 변환하기 위한 템플릿

```
(defemplate ROLELIST (multislot rolelist)
(defemplate STAGELIST (multislot stagelist)
(defemplate OBJECTLIST (multislot objectlist)
(defemplate EVENTLIST (multislot eventlist)
(defemplate STAGES (slot (stage)))
(defemplate RULE (slot (targetlist)
(slot (mode)
(slot (datatype)
(slot (WHEN)
(multislot (by)
)
```

파서에 의해서 변형된 ACPL의 내용은 실제 세션 관리와 공유 객체 관리 및 통신 관리를 위한 모듈에 전달되어서 그룹 생성에 사용된다. 이를 위해서 자바 객체 GroupManager를 생성하였고, GroupManager는 그룹 생성에 필요한 정보가 입력되면 그룹 생성을 위해서 제공되는 세션 관리, 공유 객체 관리, 통신 관리 모듈의 적절한 메소드를 호출한다. 리스트 7은 ACPL로 정의한 규칙들을 실제 자바 메소드와 연결하기 위해서 파서에 의해서 생성된 규칙들이다. 이 규칙은 JESS의 지식 베이스에 들어가서 그룹 활동 중에 발생하는 여러 사건에 대해서 적합한 규칙이 적용된다.

리스트 7. CLIPS 형태로 변경된 규칙

```
(defrule Group_RULE1
(RULE (targetlist LOCAL)(mode IN)(datatype
REQUEST_TOKEN)(WHEN Teaching)(By Student))
(STAGES (stage Teaching))
= > (printout t "Deny Request" crlf )(call
RequestHandling LOCAL IN REQUEST_TOKEN
Teaching StudentDeny_Request))
(defrule Group_RULE2
(RULE (targetlist ALL)(mode IN)(datatype
CommandTeaching)(WHEN ALL)(By Lectuer))(STAGES
(stage ALL)= > (printout t "Command by Lecture" crlf)
(call RequestHandling ALL IN CommandTeaching ALL
Lectuer Command by Lecture))
(defrule Group_RULE3
```

```
(RULE (targetlist LOCAL)(mode IN)(datatype
sound)(WHEN Teaching)(By Student Attendant))(STAGES
(stage Teaching)= > (printout t "Prohibit sound" cdf)
(call RequestHandling LOCAL IN sound Teaching Student
Attendant Prohibit sound))
```

학생은 현재 개설되어 있는 과목을 조사한 후, 원하는 과목이 진행되는 그룹에 참여해서 학습한다. 학생은 일반 학생과 청강생으로 구분되며, 일반 학생은 발언권을 가질 수 있는 대신 청강생은 발언권을 가질 수 없는 차이가 있다. 일반 학생은 그룹 활동에서 자신의 정책을 가질 수 있으며, 그룹 의장이 결정한 그룹 정책과 함께 전체적인 그룹 활동을 정의한다. 학생 A는 강사로부터 단계 변화 이벤트가 전달되는 경우 Change Mode 메시지를 발생시켜 전체 응용의 단계를 변화시킨다. 또한 짚은 시간에 학생이나 청강생이 화이트보드들 통해서 데이터를 전달하는 경우 이를 화면에 표시하지 않도록 하고 토론의 결과를 보고하는 경우에도 자신의 화이트보드에 반영하지 않으려 하는 경우 리스트와 같이 ACPL로 기술한다.

```
RULES :
  RULE : LOCAL : IN :- CommandTeaching
  WHEN ALL BY Lectuer => Change Mode
  RULE : LOCAL : IN :- whiteboard WHEN
  Question BY Student Attendant => Prohibit whiteboard
  RULE : LOCAL : IN :- whiteboard WHEN
  Report BY Student Attendant => Prohibit whiteboard
```

학생 B는 학생 A와 마찬가지로 모든 단계에서 단계 변화 이벤트가 전달되는 경우 응용의 단계를 변경한다. 또한, 짚은 시간에 학생이나 청강생으로부터 음성 데이터가 전달되는 경우 이를 전달 받지 않으려고 하는 경우 리스트와 같이 ACPL로 기술한다.

```
RULES :
  RULE : LOCAL : IN :- CommandTeaching
  WHEN ALL BY Lectuer => Change Mode
  RULE : LOCAL : IN :- sound WHEN
  QUESTION BY Student Attendant => Prohibit sound
```

이렇게 정의된 학생 A와 학생 B의 ACPL은 강사가 정의한 그룹 ACPL과 같은 방식으로 템플릿을 이용해서 CLIPS로 변환되고 지식 베이스에 저장된다. 추론엔진은 정의된 조건이 만족하는지를 검사해서 만족하는 경우 해당 메시지를 전달하

고, 해당 메시지에 연결된 자바 객체가 동작하도록 할 수 있다.

4.4 교수의 정책 변경

한번 정책이 결정된 후 새로운 정책을 추가하는 경우에는 전체 정책을 다시 컴파일하지 않고 새로운 규칙만 컴파일해서 지식 베이스에 추가하면 새로운 정책을 수행할 수 있다. 예를 들어서 그룹 단계 중에서 Teaching 단계가 아닌 경우에는 다른 사용자의 접근을 금지하는 경우에는 다음과 같이 ACPL을 추가하면 된다.

```
RULE : LOCAL : IN :- JOIN WHEN Question BY
Student Attendant =>Prohibit JOIN
RULE : LOCAL : IN :- JOIN WHEN Discussion BY
Student Attendant =>Prohibit JOIN
RULE : LOCAL : IN :- JOIN WHEN Reporting BY
Student Attendant =>Prohibit JOIN
```

또한 공유 객체에 대한 정책의 변경도 변경된 공유 객체에 대한 정책을 ACPL로 기술하여 지식 베이스에 변경하여 추가하여 주면 된다.

```
OBJECTS :
OBJECT : material Textbook1 USING_BY : Lecturer W
OBJECT : sound Stream USING_BY : Lecturer W
OBJECT : whiteboard Stream USING_BY : Lecturer W
```

4.5 이벤트 처리

ACPL로 정의된 정책이 파싱 되면 실제 응용 단계에서의 사용자의 행위를 기다리게 된다. ACPL로 기술된 정책들이 작업 메모리와 추론엔진에서 올바르게 동작하기 위해서는 응용에서 발생하는 사용자의 행위가 발생할 작업단계와 어떠한 기능을 요구하고 그 요구의 방향, 행위자의 역할이 fact로 변환된다. 이때 해당하는 규칙을 있을 경우에는 규칙에서 정의한 행위를 수행하는데 실제로 어떠한 이벤트가 발생하였는지를 검사하기 위해 자바 객체인 RequestHandling 객체를 호출한다. 호출되는 함수인 RequestHandling 자바 객체는 실제 해당기능을 수행하는 객체의 메소드를 실행시키는 역할을 담당한다. 규칙이 만족하여 RequestHandling 자바 객체를 호출할 때는 모든 규칙을 만족시킨 템플릿 내에 데이터 값을 모두 전달하여 해당하는 기능을 수행하는 메소드가 무엇인지를 판단할 수 있는 기준으로 사용되게 한

다.

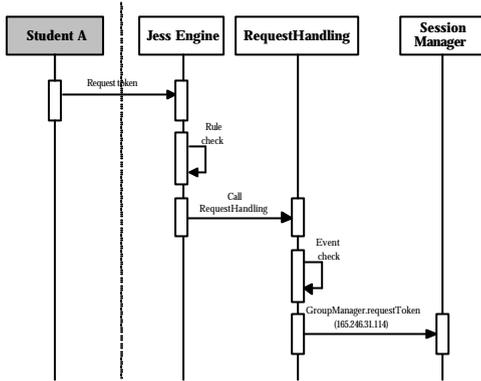


그림 1. 토큰 요구에 대한 이벤트 처리 과정

그림 1에서와 같이 학생 A가 발언권을 요구할 때 Jess 엔진에 의해 규칙이 검사되고 ACPL에서 정의한 규칙인 LOCAL : IN :- REQUEST_TOKEN WHEN Teaching BY Student에 알맞은 fact일 경우 RequestHandling 객체를 호출하며 RequestHandling 객체는 세션 관리자의 GroupManager.requestToken() 메소드를 호출하게 된다.

그림 2는 ACPL로 정의한 규칙에 위반되는 이벤트가 발생하였을 경우 이를 처리하는 과정을 보여주는 그림이다.

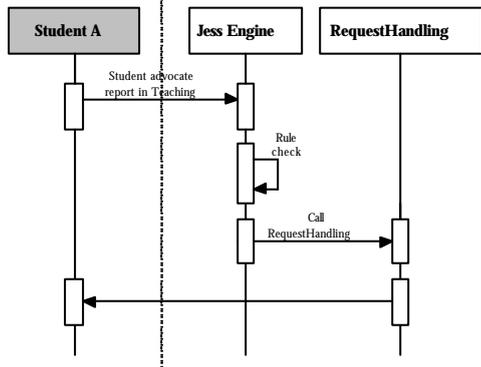


그림 2. 규칙에 위반되는 이벤트의 처리

그림 2는 학생 A가 Teaching 단계에서 레포트를 제출하는 이벤트를 처리하는 과정이다. 그러나 ACPL 정의에서 레포트는 Discussion 단계에서만 받을 수 있다는 규칙인 Lecture : IN :- ReportResult WHEN Discussion BY Student를 정의하였기 때문에 학생 A가 발생한 이벤트는

RequestHandling 객체에 의해 규칙이 위반사실을 통보 받는다.

이 실험에서는 제안한 모델을 이용하여 공동 작업의 가상시나리오에 적용하여 보았다. 제시된 특정 시나리오를 ACPL로 적절히 기술할 수 있었으며, ACPL을 CLIPS 형태로 변경할 수 있었다. CLIPS 형태로 변경된 정책은 지식 베이스에 저장되어서 추론엔진에 의해서 조건을 조사해서 조건이 만족하는 경우 특정 자바 객체의 메소드에 전달함으로써 원하는 행위를 수행할 수 있게 하였다.

따라서 기존의 공동작업 시스템과 비교해서 동적으로 정의되는 정책을 응용의 변화 없이 적용할 수 있는 공동작업의 정책을 설계할 수 있었다. 기존의 공동작업 시스템은 동적으로 변경하는 정책에 대해서 응용을 다시 개발하거나 정책을 변경해서 응용을 다시 컴파일 해야 하는 문제점이 존재했다. 따라서, 동적으로 정의되는 정책이 응용의 변화 없이 적용될 수 있는지는 기존의 공동작업 시스템과 비교 평가에서 중요한 기준이 된다.

C나 C++로 응용 프로그램 내에 정책 루틴이 구현되어 있는 경우 정책 변경이 발생하면 응용 프로그램의 정책 루틴을 다시 작성해서 컴파일 해야 한다. 그러나, 응용 구현과 정책 기술을 분리함으로써 사용자가 그룹 정책이나 개인 정책을 변경하는 경우 응용 프로그램의 수정 없이 원하는 정책을 수행하는 공동작업의 정책을 정의할 수 있었다.

V. 결론

이 논문에서는 분산 환경에서 공동 작업을 위한 적응적 정책 조정 모델을 제시하였고, 제시한 모델을 타당성을 보이기 위해 프로토타입 시스템을 구현하여 실험하였다.

제안한 정책 조정 모델은 정책 기술 언어와 추론 엔진을 묶으로써 다양한 공동 작업 응용의 개발에 유연하게 대처할 수 있으며, 단순화된 형태의 언어를 제시함으로써 일반 사용자들도 쉽게 그룹 단위 또는 개인 단위의 정책을 서술할 수 있다.

이 논문에서 제시한 모델과 응용 개발 환경의 타당성 검증을 위해 응용으로 다양한 역할과 규칙을 갖는 원격 화상 교육 응용을 개발하여, 다중 사용자 공동 작업에서 정책의 조정과 공유 객체에

대한 접근 제어가 원활히 수행되는가를 확인하였다. 이를 위해서 강사와 학생, 청강생의 역할로 구성되고 화이트보드와 화상회의 응용을 이용한 교육 환경을 구성하였다. 전체 그룹에 대한 정책과 학생 각각의 개인 정책을 ACPL로 격결하게 정의되는지를 보였으며, 정의된 ACPL이 추론 엔진에서 사용되기 위해서 CLIPS 형태로 변환되는지를 보였다. 또한, 새로운 규칙을 응용의 변화 없이 추가될 수 있음을 보였다. 따라서 제안한 모델은 그룹 정책과 개인 정책의 격응적인 대응으로 다양한 응용에 격용이 가능하고, 단순한 정책 기술 언어를 제공하여 실시간 다중 사용자 공동 작업 환경에서 유연성을 제공할 수 있다.

* 참고 문헌

- [1] W. Sharrock, J. Hughes, J. O'Brien, M. Rouncefield, T. Rodden and D. Calvey, CSCW Requirements, COMIC ESPRIT BASIC RESEARCH PROJECT 6225, 1995. 8.
- [2] 한국정보과학회, 특집 : Collaborative Computing, 제 16권, 제 7호, 1998.7.
- [3] J. J. Trevor, "Infrastructure Support for CSCW, The degree of Doctor of Philosophy, Department of Computing Lancaster University, 1994. 12.
- [4] W.K.Edwards, Coordination Infrastructure in Collaborative Systems, A Dissertation Degree of Doctor of Computer Science, Georgia Institute of Technology, Nov. 22, 1995.
- [5] W. J. Tolone, INTROSPECT: A Meta-level Specification Framework for Dynamic, Evolvable Collaboration Support, A Dissertation Degree of Doctor of Computer Science, University of Illinois at Urbana-Champagin, 1996.
- [6] F. DeCindio, G. DeMichelis, C. Vassallo and A. M. Zanaboni, "CHAOS as Coordination Technology," Proceedings of the conference on CSCW'86, pp.325-342, 1986.
- [7] T. Kreifelts, U. Pankoke-Babatz and F. Victor, "A Model for the Coordination of Cooperation Activities," Proceedings of the International Workshop on CSCW, Berlin, pp.85-100, 1991.
- [8] Greenberg, S. <http://www.cpsc.ucalgary.ca/group/lab/papers/index.html>
- [9] The MASH Research Group, <http://mash.cs.berkeley.edu/mash/pubs/index.html>
- [10] ISO/IEC, "Digital Storage Media Command & Control," ISO/IEC JTC1/SC29/WG11 N1100 rev1, p.5, Nov.1995
- [11] CLIPS. <http://www.gbg.net/clips/CLIPS.html>
- [12] Ernest J. Friedman-Hill, "Jess, The Java Expert System Shell," <http://herzberg.ca.sandia.gov/jess/README.html>, 1998

이세훈



1985년 인하대학교 전자계산학과 졸업(이학사)
 1987년 인하대학교 대학원 전자계산학과 졸업(이학석사)
 1996년 인하대학교 대학원 전자계산공학과 졸업(공학박사)
 1987~1990년 해병대 전산실 분석장교
 1991~1993년 비트컴퓨터 기술연구소 선임연구원
 1999~현재 (주)토마토아이 기술고문
 1999.6 한국멀티미디어기술사
 1993~현재 인하공업전문대학 전자계산학과 부교수
 관심분야 : 분산적체컴퓨팅, 멀티미디어, 소프트웨어공학, 원격교육

* 본 연구는 인하공업전문대학 산업기술연구소 99년도 교내 연구비 지원에 의하여 수행되었음.

백영태



1989년 인하대학교 전자계산학과 졸업
 1993년 인하대학교 대학원 전자계산학과 졸업(공학석사)
 1993년~1998.2 대상정보기술(주) 정보통신연구소 선임연구원
 1998.3~현재 김포대학 컴퓨터계열 멀티미디어전공 전임강사
 관심분야 : 지능형 교육 시스템, 멀티미디어 저작도구, 웹컴퓨팅

서 대우



1992년 인하대학교 전자계산학과 졸업
 1994년 인하대학교 대학원 전자계산학과 졸업(공학석사)
 1994년~1997년 대상정보기술연구소 연구원
 1998~현재 (주)사이맥스 기술연구소 연구실장
 관심분야 : 가상현실, 컴퓨터 애니메이션, 멀티미디어 저작도구