Dynamic Coordination Policy for Cooperative Learning System

YeongTae Baek
Dept. of Multimedia, Kimpo College, KOREA
hannae@kimpo.ac.kr

SeiHoon Lee
Dept. of Computer Engineering, Inha Technical College, KOREA
seihoon@true.inhatc.ac.kr

Kyungseob Yoon
Dept. of Computer and Information System, Inha Technical College, KOREA
ksyoon@true.inhatc.ac.kr

ChangJong Wang
Dept. of Computer&Science and Engineering, InhaUniversity, KOREA
cjwang@selab.inha.ac.kr

Abstract: Most of these cooperative learning system has a static type that the learning policy decided the beginning of system development. There is a problem that these static policy model can't change dynamically when people perform learning system. Because it is possible to static modification, it includes problem that every policy for cooperative learning should be decided before the execution of learning system. This prevents performing of original facility of cooperative learning. In this study, we proposed the model of coordination policy that can change policy dynamically through separating of coordination and implementation.

1. Introduction

Through the improvement of Computer system capabilities and extension of information network communication system, it is increased to request of Computer Supported Cooperative Work (CSCW) applications which support that a number of people globally distributed make a decision and do their cooperative work through the sharing their information. Especially, research about cooperative learning to get educative effectiveness is being actively performed.

Most of these cooperative learning system has a static type that the learning policy decided the beginning of system development. There is a problem that these static policy model can't change dynamically when people perform learning system. Because it is possible to static modification, it includes problem that every policy for cooperative learning should be decided before the execution of learning system. This prevents performing of original facility of cooperative learning.

In this study, therefore, we propose the model of coordination policy that can change policy dynamically through separating of coordination and implementation. The proposed coordination policy model provides interaction structure which import two concept of virtual bus to cooperative learning system developers and defines CPDL(Coordination Policy Definition Language) to model coordination policy. Also, using rule base method not procedural method to control and process defined policy with CPDL, it is possible to process flexibly about changing status dynamically.

2. Coordination Policy

It is a aim that cooperative learning system easily provide works for learning happen to between many learners using computer. Cooperative learning is coordinated by special policy. In other words, cooperative learning can restrict role and interaction of participating learners. So, in order that cooperative learning system support group learning effectively, it should include these rules in software. These rules are coordination policies. Coordination policy include access control, concurrency control, floor control, role/session constraints, exception handling.

Coordination policy can be made by working type and organization structure of learning groups, and that can be coordinated by other policy according to learning stage. The development of cooperative learning system is made by communication of experts, developers, or users for defining of coordination policy generally. These method are defined when system is developed, these policy is called once-and-for-all policy which is applied every case. A lot of research is being progressed about very various cooperative learning system in CSCW during past 10 years. Most of these system developed by once-and-for-all policy. These approaches don't provide flexible cooperative learning because the decided policy on system development applies to every case. So, it is required development of cooperative learning system which import method separates system and policy.

3. Dynamic Coordination Policy

In this chapter, we propose coordination policy model. A necessary coordination policy on overall cooperative learning is separated learning group policy and learner policy in proposed model. And it is possible that dynamic policy changing through separating system implementation

3.1 CPDL

CPDL is a language to define learning group policy and learner policy. It is used to define policy about overall cooperative learning. As developer of cooperative learning system defines policy with CPDL besides general application such as white board or tele-conferencing, he can provide cooperative learning applications with new policy to learner without modification of application in spite of dynamic change. Policy with CPDL consist of part about type of learning group, definition of sharing application to use in learning group, definition of necessary events in learning group and each rules.

Policy about learning group type

The type of learning group consists of title of learning group, its role, stage of learning group, list of application, number of learners, openness of learning group, control method of right to speak, sorts of events, and rules of learning group. Role of learning group and stage of learning group operation is defined [list 3-1]. Objects and events used group can composed of object definition list and event definition list. And numbers of learner which can participate in group are able to be defined overall number of learners and each role base number of learner. Group property has an only one property either PUBLIC or PROTECTED. If group property is PROTECTED, group can restrict learners participate in group using secret numbers. The control of right to speak among learners in group are TOKEN_CONTROL, ROUND-ROBIN, FREE method.

[List 3-1] Group 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_defintionlist | nothing
event_list ::= EVENTS : event_definition | nothing
id_list ::= identifier, id_list | identifier
object_definitionlist ::= object_definition | object_definition object_defintionlist
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

Objects and event definition

Group can share object and it can different from sharing application according to role. Also it can use object to different right though it shares same object. Object definition consists of object name, class name, role, and right used object.

[List 3-2] Object Grammar
object_definition ::= OBJECT : objectname classname objectactivity right
objectname ::= NAME : identifier
classname ::= CLASS : identifier
objectactivity ::= USING_BY : identifier
right ::= W | R

Events happen to interaction process of learners participated in group. And appropriate processes of events need to support smooth learning activity. The events type is consisted of events source, events destination, and field list consisting of events.

[List 3-3] Event Grammar
event_definition ::= EVENT identifier : simple_event
simple_event ::= event_src event_des event_fie ldlist
event_src ::= SRC : context_list
event_des ::= DST : context_list
event_fieldlist ::= FIELDS : fieldlist | nothing

fieldlist ::= FIELD : fieldlist | nothing

Rule definition for group activity

It defines that how interaction of learners happen to learning group and pass event of learners process through rules. The rules consist of condition and behavior list. If the condition is satisfied, the descriptive behavior is performed to appropriate learner. In the case of learner, the target can be restricted to itself, and can be the learner performing the specific roles and all learners. The condition is appropriate to the case of input or output of specific data.

[List 3-4] Rules using group
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

3.2 Two way coordination policy

Learners participated in cooperative learning has common topic. So, coordination policy about group affects on interaction among all participants. Also, learners interact with other people according to purpose and policy of itself. However, because legacy study defined only group coordination policy, individual policies of each learner don't be considered.

In this study, it support both group coordination policy and individual policies of each learner, coordination policy of overall cooperative learning perform through harmony of two coordination policy. To this, the bus concept is imported. The bus consists of collaboration bus which affect on learning group coordination policy and coordination bus which affect on learner's policy. Because learners participating group is connected with collaboration bus, it is controlled by collaboration bus learning group coordination policy. Coordination bus is connected between cooperative learning applications like whiteboard and tele-conferencing and private coordinator for controlling learner policy. So, because each application is connected Coordination bus, it is controlled by Coordination bus learner coordination policy.

learning group coordination policy

Learning group policy is defined with CPDL by group creator. Group policy defined by group creator apply to every learner participated in group and use the basic policy of group activity. So, it has higher priority than learner policy and is valid as long as a group exists. The following is items in group coordination policy.

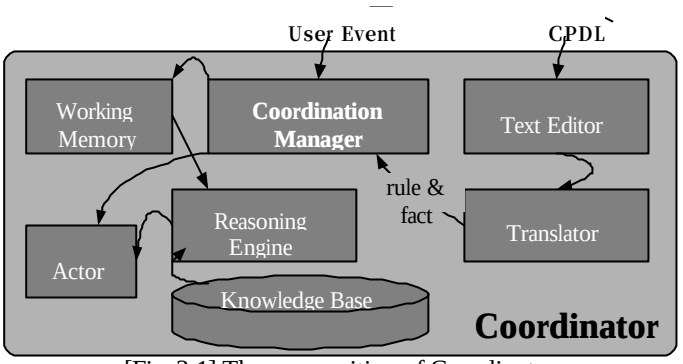
- *Group Name*
- *used Role in group*
- *used shared object in group*
- *the number of all learner and learner per role participated in group*
- *property of group*
- *group title*
- *type of right to speak*
- *a way of processing when the defined events are generated*

Learner coordination policy

Learner policy defines rule about transferred data by system. So, Most of learner coordination policy is rule definition.

3.3 Coordinator

Cooperative learning policy is descripted by CPDL and it is managed and is executed by Coordinator. Composition of Coordinator is illustrated in [Fig. 3-1]. Coordinator consists of text editor, translator, coordination manager, working memory, reasoning engine, knowledge base and actor. Text editor is a tool that learner can describe with CPDL. Translator is a parser that convert CPDL into coordination grammar type. Coordination manager controls policies of overall cooperative learning.



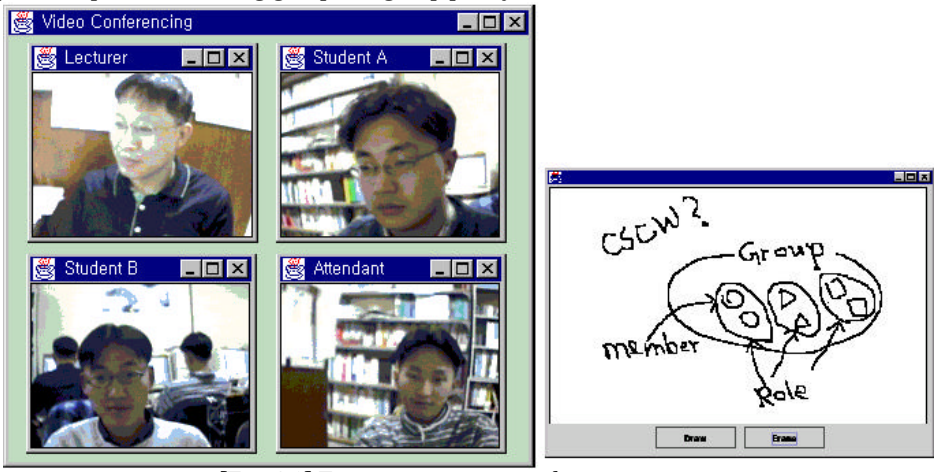
[Fig. 3-1] The composition of Coordinator

4. Experiments

we implement cooperative learning system to show propriety of the proposed dynamic coordination policy. And we show that possible interaction of users in group according to learning group policy and learner policy through processing of implemented system.

4.1 Development environment and cooperative learning system

Experiment environment consist of one teacher, two learners, and one attendance, and in this class, teacher decide the type of cooperative learning group and group policy.



[Fig. 4-1] Executing cooperative learning system

[Fig. 4-1] show executing windows. The overall cooperative learning system consist of white board and video-conferencing using real time stream. And, audio/video data of system and image data of whiteboard is controlled flowing according to group policy. The cooperative learning system operates on Microsoft Windows NT 4.0 and Windows 98 in Pentium PC. To communicate with client and server, we use TCP/IP protocol in 100base-TX Ethernet. Also, the following describes implement environment.

Developing Tools : Java Development Kit 1.x, Real time Movie : software encoding/decoding module of MPEG-1

To do cooperative learning, first teacher decide cooperative learning group policy. After that, if group is created, learners and attendance participate in group, and decide their policy. The necessary information to create group consist of group name , roles to use group, working steps of group, number of learners and role to participate group, information of objects to be used by learners of group, control type of right to speak, group property, definition of event to use group, and rules of group activity. These group type and group policy is described in [List 4-1] using designed CPDL in this study.

[List 4 -1] Group policy of education system
Classroom : "Overview of CSCW"
ROLES : Lecturer Student Attendant
STAGES : Teaching Question Discussion Reporting
.....
EVENTS :
EVENT CommandTeaching

```
    SRC : Lecturer
    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 :
      Lecture : IN :- ReportResult WHEN Discussion BY Student
         => "Report Result "
    RULE :
      ALL : IN : - CommandTeaching WHEN ALL BY Lecturer
         => "Command by Lecture"
    RULE :
      LOCAL : IN :- sound WHEN Teaching BY Student Attendant
           =>"Prohibit Speech"
```

In the case of lecture step, group rules ignore request of right to speak of learners, and in the case of discussion step, if learners try to pass to teacher discussion result, it describe CPDL which can receive discussion result. Also, during lecture, teacher can pass image data from learner or attendance, but data trans mission through audio data and whiteboard restrict for smooth lecture.

4.2 Group creation

CPDL described in [List 4-1] change to necessary information for group creation through parser. The necessary information for group creation consists of rules to be used group policy management as saving knowledge base. The Parser uses template such as [List 4-2] to change CPDL type to CLIPS type.

[리스트 4-2] Template for changing CPDL to CLIPS

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

The contents of CPDL is changed by parser is transmitted manager handling a portion and then, it used. [List 4-3] are rules to connect definition rules with CPDL with java method. This rule becomes of knowledge base of Jess, it is applied appropriate rule about several accident happen during group activity.

[List 4 -3] The changed rule of CLIPS type

```
(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 Student
     Deny_Request))
(defrule Group_RULE2
  (RULE (targetlist ALL)(mode IN)(datatype CommandTeaching)
      (WHEN ALL)(By Lecturer))
  (STAGES (stage ALL)
  = > (printout t "Command by Lecture" crlf)
    (call RequestHandling ALL IN CommandTeaching ALL Lecturer 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" crlf)
    (call RequestHandling LOCAL IN sound Teaching Student Attendant Prohibit sound))
```

4.3 Participation of learner

Learner can owns policy on group activity and defines owns group activity in addition to group policy defined by group chairman. When receiving change event from teacher, Learner-A's System make the overall application step change by generating "Change Mode" message. Also, if received data from learner or attendant in question time, System should not show the received data and if don't show the received data on whiteboard in reporting the result of conversation, learner-A describes like the following list.

```
RULES :
    RULE :
      LOCAL : IN :- CommandTeaching WHEN ALL BY Lecturer
```

```
=> "Change Mode"
   RULE :
    LOCAL : IN :- whiteboard WHEN Question BY Student Attendant
=> "Prohibit whiteboard"
   RULE :
    LOCAL : IN :- whiteboard WHEN Report BY Student Attendant
=> "Prohibit whiteboard"
```

Learner-B like learner-A changes the application step on receiving step change event in overall step. Also, if received the voice data from learner or attendant in question time, it describes with CPDL like the following list for no receiving the receiving data.

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

CPDL of learner A and B convert CLIPS using template like group CPDL and it saves in Knowledge Base. If the condition is satisfied, Reasoning Engine transmits message, and makes it launch java object connected message.

4.4 Policy change of teacher

In the case of adding new policy after it decided policy, System can execute new policy just compiling new rules and adding to knowledge base without recompiling overall policy. For example, if group stage is not "Teaching" group stage, and try to prohibit access of other users, it add the following CPDL

```
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"
```

5. Conclusion

In this study, we proposed the model of coordination policy that can change policy dynamically through separating of coordination and implementation. The proposed coordination policy model provided interaction structure which import two concept of virtual bus to cooperative learning system developers and defined CPDL(Coordination Policy Definition Language) to model coordination policy. Also, using rule base method not procedural method to control and process defined policy with CPDL, it was possible to process flexibly about changing status dynamically. The proposed coordination policy model solves problem of static policy model, which it can't change policies dynamically on the execution of learning system, and can provide various learning step and learning way through changing coordination policy dynamically.

6. References

[ACM98] ACM, Proceedings of computer Supported Cooperative Work, 1998

[Blu95] C. Blum, P. Dubois and R. Molva, "A Semi-Distributed Platform for the Support of CSCW Application," Proceedings of the 1st International Distributed Conference IDC'95, 1995.

[Edw95] W.K.Edwards, Coordination Infrastructure in Collaborative Systems, A Dissertation Degree of Doctor of Computer Science, Georgia Institute of Technology, Nov. 22, 1995.

[FER95] D. F. Ferraiolo, J. Cugini and R.Kuhn, "Role-Based Access Control:Features and Motivations," National Institute of Standards and Technology, August, 1998.

[GUS96] M. Gustafsson, N. Shahmehi, "A Role Description Framework and its Applications to Role-Based Access control," Proceedings of IEEE Fifth Workshops on Enabling Technology: Infrastructure for Collaborative Enterprises(WETICE' 96), June, 1996.

[Kos98] A.P.Kosoresow, G.E.Kaiser, "Using Agents to Enable Collaborative Work", IEEE Internet Computing, Vol. 2, No. 4., pp.85-87, July-Auguest 1998

[Sik97] K.Sikkel, "A Group-based Authorization Model for Cooperative Systems", Proceedings of European Conference on Computer-Supported Cooperative Work (ECSCW '97), Lancaster, September 1997.