

## **Constructing collaborative learning activities for distance CAL systems**

C-K. Chang & G-D. Chen  
National Central University, Taiwan R.O.C.

**Abstract** Many existing Computer Assisted Learning (CAL) systems use computer programs to simulate a teacher for individual learning. However, regardless of how good the CAL system is, a student should also interact with learning companions and human teachers. Due to the popularity and ease of access to computer networks and communication tools, network facilities can be incorporated into a CAL system to make it a distance CAL system. To attain an effective learning outcome, a distance CAL system must be able to: define a group learning model; constrain and advise the students to learn according to the model, and have query facilities which provide the teacher with information about what a student has done and learned, and what a student should do next. A rule-based system that is conventionally used cannot sufficiently provide those required functions. This paper suggests why this is the case and how an active database system can achieve those goals.

**Keywords:** Active database; Distance CAL; Learning activity; Production rules; Rule-based system

### **Introduction**

Students in a distance CAL system can learn by discussing, asking questions, or collaborating through a computer network. Wells (1992) summarises the feasibility of constructing various learning activities through conventional network tools for distance education. Hiltz (1994) proposed the notion of the 'virtual classroom' for distance learning. The functionality of the virtual classroom includes conferences, messages, notebooks, document access, blackboards, examinations, gradebooks and response activities. A distance CAL system on the computer network is a virtual classroom that re-creates the range of activities of a conventional classroom. However, to achieve a good learning outcome, a teacher needs to organise the group learning activities and understand the behaviour of students well.

Collaborative learning (Bricker *et al.*, 1995; Monaghan & Clement, 1995) styles are used in conventional classrooms. The students can be divided into groups

---

Accepted: 20 July 1996

Correspondence: Gwo-Dong Chen, Department of Computer Science and Information Engineering, National Central University, Chung-Li 320, Taiwan R.O.C.  
Email: chen@db.csie.ncu.edu.tw

**2**

and the members of a group can be assigned different roles. Tasks (homework or exercises) can be divided into sub-tasks that can be assigned to group members with different roles to tackle. Concepts can be discussed from different perspectives by each member with his/her specific role. The learning effect can be improved because students can observe and learn things from different views. Additionally, students can be motivated through group and peer pressure and may be more willing to cooperate, exchange ideas, and share experiences in order to get a good score for their group activity. Many formats for group interaction are frequently used to stimulate learners to be active, e.g. students as teachers, role playing, formal debating, writing groups and team projects. Collaborative learning is achieved by activities through different kinds of group interaction.

Existing communication tools provide ways for users to communicate and share information on a network, such as Internet Relay Chat (IRC), Bulletin Board Systems (BBS) and World Wide Web (WWW). These tools unfortunately treat every user as having the same role. Although, information in BBS and News system may be organised according to different interest groups, users are free to choose any group to post and read information. The users do not have any responsibility to do anything. However, in order to learn, a student must be directed or motivated to do something that is good for his/her learning. If generic network tools are to be used for learning, the teacher should participate in the discussion and assign roles, notify students, enforce schedules, guide students to what he/she should do, and accumulate information to understand the performance of students. These are all onerous tasks for the teacher.

The basic requirements for supporting a collaborative learning activity are rules/protocols, task agenda and interaction transcript. Most existing distance CAL systems construct learning activities by using existing network tools which do not have such functionality. For instance, a teacher wanting to construct a collaborative learning activity on the computer network must implement mechanisms for regulation, task setting and transcript record keeping on the computer network. Otherwise, the teacher should manually handle these tasks by himself/herself. A number of difficulties are encountered in constructing collaborative learning activities in current distance CAL systems.

First, the teacher has the difficulty of constructing a collaborative learning activity because most network tools do not support a collaborative style of working. The teacher must arrange the style of collaborative work and define what a group member should do. For instance, a teacher issues a homework that should be a collaborative task and this homework is divided into four parts. Each part is assigned to a particular group member and each group has a leader who should combine the four parts into a complete solution. For each part, the assigned member can present his idea and solution and other members can only provide suggestions and arguments. However, the group leader can modify the sub-solution. Group members can only view the results and discussion of their group once they have contributed their part. These issues are referred to as the *specification and regulation* problem.

Second, the teacher must maintain the progress of an activity because

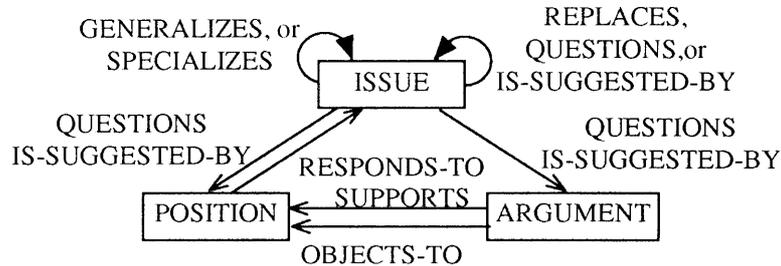
network tools cannot automate the agenda of an activity. Take the example introduced above: the teacher must define a schedule for each group, for each sub-part and for the final outcome. Also, if members have an opinion regarding a sub-part idea or solution, the assigned leader must respond within 24 hours. This issue is called the *automating schedule* problem.

Third, a teacher wanting to know what is going on in collaborative learning activities must make a great effort to find the necessary information in the large volume of generated data. As proposed by Davie (1987), the historical data of a collaborative learning activity are called a 'complete *transcript* of the interactions' and in monitoring the progress of activities a teacher will become overloaded with information (Hiltz & Turoff, 1985). Furthermore, a teacher cannot dynamically prepare a course without the mechanism of querying the transcripts, which represent the students' learning history. The teacher may want to identify the member who delays a group result or know the complete learning history of a student. This issue may be called the *recording and querying transcript* problem.

This paper attempts not only to resolve the above mentioned problems of distance CAL systems from a database perspective, but also to identify the required functionality for constructing collaborative learning activities in a distance CAL environment. The initial motivation of using a database system originates from considering the *transcript recording and querying* problem. A survey of database technology reveals that an active database may solve the *specification and regulation* and *automating agenda* problems. To resolve the specification and regulation problem, the active database (McCarthy & Dayal, 1989) is applied to construct a collaborative learning activity, which is independent of the network tools being used. To resolve the automating agenda problem, the agenda of an activity can be automated by integrating temporal issues in an active database. To resolve the transcript recording and querying problem, the query powers of the database can solve the problem of information overload and make it feasible to query the transcript of an activity. Advanced technology (active database) and theory (temporal logic) are employed to draft the infrastructure of distance CAL systems. Finally, an illustrative example demonstrates the effectiveness of the proposed approach.

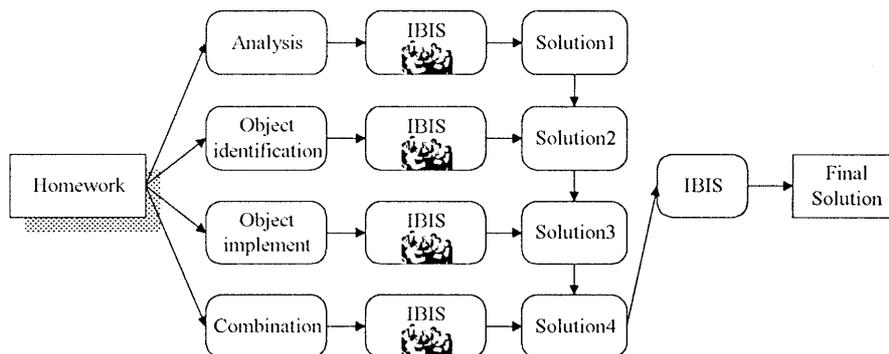
### **A collaborative homework example**

A illustrative example is presented here to elucidate the difficulties of constructing collaborative learning activities. The Issue Based Information System (IBIS) is a discussion model for collaborative learning activity (Kunz & Rittel, 1970). The IBIS model, illustrated in Fig. 1, starts with a problem or issue. Many positions are then proposed to resolve the issue and many arguments are available, which may support or object to a position. A position may derive a new issue, which may generalise, specialise, or replace the previous issue. The IBIS model is effective in collaboratively resolving a design problem. Conklin and Begeman (1989) developed a graphic interface to represent the IBIS model in hypertext style.



**Fig. 1.** The IBIS model

It is assumed that a teacher would construct a collaborative homework activity by using the IBIS model. The teacher wants students to apply four steps (i.e. analysis, object identification, object implementation, combination) for a homework on object-oriented programming. Students are subsequently divided into four groups and each group is assigned to discuss one of the steps of the IBIS model (see Fig. 2). Moreover, there must be a moderator in each group to summarise a solution before the task deadline.



**Fig. 2.** A collaborative homework task represented by the IBIS model.

A student can propose and defend a position for his/her group; however, he/she can only propose the argument for the other group. A discussion for reflection is held in the IBIS model after the combination step. Solutions of the four steps must be refined if some problems are found in the discussion for reflection. A final solution is summarised if the discussion for reflection does not find any problem.

The teacher has a number of tasks. First, students must be divided into four groups and each group has a member assigned as a moderator, who can determine whether to derive a new issue and summarise a solution. Defining the rules for the behaviour of each member is necessary. Second, a moderator should notify the owner of a position to defend his/her position when other members argue the position. An active mechanism to reduce the moderator's task is required to make notifications automatic.

**Client**

Role: student

Capability:

- propose a position to an issue in his/her group.
- propose an argument to a position in any group.
- request the moderator to derive a new issue.
- answer objects-to arguments of proposed position.

Constraint:

- When requesting a teacher to derive a new issue from a position:
  - if the student did not propose a position
  - then deny and notify the student 'propose your position!'

Role: moderator

Capability:

- determine the initial issue.
- determine whether to derive a new issue.

Constraint:

- when a student requests to derive a new issue from a position:
  - if the request is not answered currently
  - then the request will be answered in the future eventually.

**Role-based activity server**

- check the identification, including group and role, of a client
- retrieve capabilities and constraints of client from active database
- receive the action from active database to notify the client

IBIS model

- when adding a new node
  - then the type of node is one of {issue, position, argument}
- when creating a link
  - then the type of link is one of {supports, objects-to, responds-to, questions is-suggested-by, generalises, specialises, replaces}
- when adding a child node to an issue
  - if the type of link is responds-to
  - then the type of the child node is position
- when adding a child node to an issue
  - if the type of link is generalises, specialises, replaces, or questions is-suggested-by
  - then the type of the child node is issue
- when adding a child node to a position
  - if the type of link is questions is-suggested-by
  - then the type of the child node is issue
- when adding a child node to a position
  - if the type of link is supports or objects-to
  - then the type of the child node is argument
- when adding a child node to an argument
  - if the type of the child node is questions is-suggested-by
  - then the type of the child node is issue

Active notification:

- when a question to a moderator for deriving a new issue is not answered
  - if the question is not answered more than two days
  - then immediately notify the moderator to make a decision.
- when an objects-to argument of a position is not answered
  - if the argument is not answered more than two days
  - then notify the owner of the position to answer the argument.
- when the deadline of a step is due
  - if the solution of the step is not summarised
  - then notify the moderator of the step to summarise a solution.

Third, the teacher may want to identify the most contentious position or the most active student from the records of discussion and so requires an effective approach to query the transcript of a collaborative learning activity. However, existing distance learning systems on the network do not have tools to satisfy these requirements. Restated, the network tools can not provide a model for group interaction.

The following specifications illustrate how to construct the collaborative homework example by a methodology from a database perspective. To depict the enhancement, an informal representation is used:

WHEN the *event* is detected and IF the *condition* is satisfied,  
THEN the corresponding *action* is performed.

The *events* relevant for an activity include: database events (insertion, deletion, modification, retrieval), and temporal events (at a given time, or at any time after a specific event), such as 'deadline of a homework'. *Conditions* mean the expected results from a database query. *Actions* imply sequences of commands that can be a database operation or external (user defined) processes, such as notification. The following specifications specify moderator, student, and server for the collaborative homework and the required infrastructure to support the specifications is given above.

### **Reasons for using an active database**

A large number of tools, e.g. Mail, Internet Relay Chat, World-Wide Web (Berners-Lee *et al.*, 1994), on the Internet provide various communication styles. Various advantages come with these tools including multimedia communication, asynchronous learning and discussion, on-line learning materials, social relationships, etc. The flexible and distributive characteristics of these tools meet some of the requirements of distance CAL.

However, distance CAL systems require some functionalities that are not available through the existing network tools, e.g. controlling access to information repositories, scheduling, inferring, and notification. Integrating network tools with advanced database technology can provide some required mechanisms that are difficult to be implemented merely by the network tools themselves. For instance, advanced database technology can control a collaborative learning activity according to the assigned regulations. Consequently, the system can automatically handle circumstance that violate the regulations.

An active database can satisfy the requirements of the collaborative homework example in the following ways:

#### *Constructing activity:*

- refers to constructing the regulations for group interaction, among which includes the roles of users, their interrelationships, and notification. For instance,
  - a moderator can derive a new issue and summarise a solution,
  - students can only propose positions in his/her own group; (1)
  - system must construct the relationships of group. (2)

*Automating agenda:*

- whether an interested event occurs and a response to it made must be monitored by an automatic procedure. For instance, the system should automatically notify a moderator who did not summarise a solution after the deadline. (3)

*Querying transcript:*

- a teacher can manage the activity through querying the transcript of an activity. For instance, list the students who often propose argument to other groups.

To emphasise the complexity of these problems, they should be contrasted with distance CAL systems using existing network tools. Most of the current distance CAL environments emphasise their *dimension of communication*, where as many communication tools are provided as possible (for instance, e-mail, mail list, WWW, and virtual place). These systems improved the courseware organisation (Ibrahim, 1994; Montague & Knirk, 1993), or pedagogical strategy (Wray *et al.*, 1994) for distance CAL systems. However, they did not consider how to construct a learning activity with the support of a database. Consequently, the requirements that can be easily achieved by a database can still not be satisfied by the existing distance CAL systems. Consequently, the proposed methodology not only supports constructing collaborative learning activities for existing distance learning environments, but also proposes a general infrastructure for constructing collaborative learning activities in a distance CAL system.

**Active database support for distance CAL systems**

An active database monitors situations of interest and, when they occur, triggers an appropriate response in a timely manner (Kim, 1995). There are three components in an active database. First, an event monitoring sub-system monitors situations of interest. Second, a set of rules can be triggered by a series of events. Third, an execution model provides the temporal aspect of a response. Table 1 indicates the requirements for attaining these goals and the relationship between the requirements and an active database.

**Table 1.** The relationship between goals, requirements and active database

<b>Requirements</b>	<b>Reasons</b>	<b>Active database</b>
Constructing activity	Define regulation Determine agenda	A set of rules, frequently called a rule base
Automating agenda	Detect event Activate action	A sub-system for monitoring events
Querying transcript	Recording action Temporal query	An execution model

An active database can satisfy the requirements because it can attain the required goals through three components:

*Constructing activity:*

Monitoring activities is essential in a distance CAL system. To effectively supervise students' actions, a distance CAL system must derive users' interrelationships and attach constraints to communication behaviours.

- *Assign Role* includes specifying the capabilities and constraints of a user, so the users mentioned in (1) can be defined as:
  - moderator's capability: derive a new issue and summarise a solution  
(1a)
  - student's constraint: propose positions in his/her own group  
(1b)
- *Determine agenda* refers to the workflow of an activity. For instance, a moderator should make a summary after a conference activity. (2) can be refined as:
  - Rule 1 for moderator: summarise a solution when a deadline is due.  
(2a)

*Automating agenda:*

This component includes two mechanisms: first, an active mechanism for checking conditions determines whether the event occurred; second, a language specifies the responding procedure with temporal issues.

- *Detect event* is based on the determined agenda, an active mechanism monitors various events, e.g. the data access, and database update. The actions of an agenda specification will be triggered when an event satisfies the conditions of an agenda specification. In (3), it refers to
  - System should automatically do something when Rule 1 is triggered. (3a)
- Activate action which what to do when the event is triggered. The action specifications are associated with roles or activities. For instance, the specification in (3) is:
  - Notify the moderator to summarise a solution (3b)

Furthermore, an active database system is a superset of a traditional database system, that can naturally provide the power to query the transcript of an activity by a query language, e.g. SQL. The other advantages, related to constructing distance CAL systems, of a DataBase Management System (DBMS) include: controlling redundancy, sharing of data, restricting unauthorised access, representing complex relationships, enforcing integrity constraints, and reducing application development time. However, problems arise when using active databases to construct a learning activity.

**Active database systems**

Active database systems are defined as database systems with production rules (Brownston *et al.*, 1985). The facility of production rules provides a conventional database with some advanced features, such as triggers, alerts, protection, and version control. Most importantly, active database systems can provide functionality to support the requirements of a collaborative learning activity. First, active database systems can provide the *event-based interaction*, which

serves as an inference engine matching the conditional parts of agenda rules (2a). Second, an active database system can attach *collaborative constraints* to a role which maintains the relationships of users' roles. Third, the execution model can integrate learning activity *with temporal triggers*, which guarantee the workflow of an activity.

#### Event-based interaction support for automating agenda

An activity must be able to automatically inform participants when certain situations arise. To capture the significant situations, the infrastructure of an activity should provide entry, called *event*, to represent the situations and a mechanism, called *match*, to detect whether the situations occur. When the specified situations are detected, a system will actively respond. The functionality is called event-based interaction in active database systems. In example (3a), the part of event is:

when *event*; where *event* is 'Rule1 is triggered' (3a.1)

and the part of the match is:

System should automatically . . . (3a.2)

The active mechanism based on event-based interaction is a more intuitive approach than the conventional state-based interaction using shared blackboards to effectively control the agenda of a collaborative learning activity. Restated, the event-based interaction avoids predicating users' behaviours in distance CAL systems. Instead, the sequence of events may imply pedagogic issues in distance CAL systems. Recording the sequence of various types of events becomes important. For instance, the event of retrieving taught contents may indicate a reflection. Techniques recently developed in the area of active database use can be efficiently adapted to provide this capability (Sistla & Wolfson, 1995).

#### Production rule support for assigning role

Assigning role implies that a user should obey the constraints of the assigned role for collaboration. The constraints of a role for collaboration refer to the legal actions that a person can do in a collaborative learning activity. Collaboration is more than communication because it requires sharing common goals to achieve tasks. Besides sharing goals, participants of a collaborative learning activity must comply in using some interaction protocols for collaboration, which is accomplished by assigning the role in this approach. For instance, a moderator's role differs from others in (2a). Furthermore, to make the activity progress on schedule, all roles must comply with the moderator's decisions on the activity. Collaboration constraints force users to comply with the interaction protocol. Production rules of the active database provide a mechanism for elegantly supporting the collaborative constraints of a role. A production rule is of the form:

condition → action

The condition part of a production rule determines the action's legal role. The

action part of a production rule indicates the possible behaviour. In example (2a), one of the production rules is:

is a moderator → make a summary when the deadline of a step is due

Consequently, an active database provides a declarative specification to define the users' collaborative relationship.

#### Temporal trigger support for determining the agenda

Temporal triggers refer to adding the temporal issues in the parts of event and condition. Temporal relationships play a considerable role in distance CAL environment. Accurately describing the time dependency of an activity requires a mechanism to support determining the agenda of an activity. Besides, there is also a need for explicit representation of time, and processing some cases under time-constraints. For instance, a moderator may determine the time and duration to hold a conference. Furthermore, the transcript of an activity must be recorded with a data type of time because an educator requires this information for behaviour analysis.

Most existing distance CAL systems are static in describing the agenda of an activity. Restated, only 'current' data are reserved. However, changes in the real world are traditionally reflected by updating the database which results in losing old information. The execution model of an active database can resolve the problem and support the temporal trigger in the following ways. First, the transcript of the execution model can be stored permanently because an activity may be asynchronous and last for a long period. The transcript of the execution model includes factors such as when an event was triggered, who activated the event, and the sequence of event-based interaction. Second, the current state of activities can be described in an active database and a temporal trigger will automatically respond. Third, an activity can be reused to create a new activity because the agenda is determined according to the execution model, which is a sequence of database states.

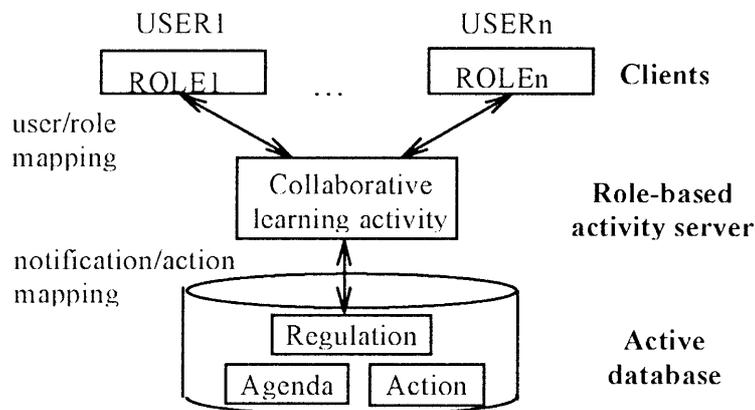
To sum up, the approach integrates the model of an active database with temporal issues. In the next session, the operation of the approach and an informal language to specify an activity are suggested. The informal language, incorporating temporal issues, is similar to the specification language used in active database literature. Consequently an end user, teacher, can construct a collaborative learning activity on a formal basis.

#### **Procedures for using an active database**

Some assumptions are needed regarding the procedures for using an active database to construct collaborative learning activity in distance CAL systems. First, the regulations of a collaborative learning activity are based upon the realistic assumption that users participating in a collaborative learning activity are not autonomous. Detecting whether a role's tasks was completed is subsequently necessary because of this assumption. Second, the collaborative learning activity must be accessible at a distance and so there must be a server that users can connect to from any place. Third, users should be able to

communicate and exchange data with the server and so there must be a user client program to interact with the user and communicate with the server.

Based on the above assumptions, the collaborative learning activity consists of *clients*, *role-based activity server*, and an *active database*. The client is responsible for assigning role and related abilities to users by translating the message from the active database, communicating with server, and generating alarms to inform users to do something. The role-based activity server is introduced to transfer users' messages into the active database, so that it can maintain and support the role of each user. The active database is responsible for filtering communication messages, assigning temporal constraints for activities, and determining users' legal behaviours in an activity. Figure 3 illustrates the necessary software architecture for these procedures.



**Fig. 3.** The necessary software architecture

### Enhancing an information server to a role-based activity server

Although the software architecture required was enhanced, it is compatible with existing information systems, e.g. World Wide Web, Gopher, and Internet Relay Chat. Furthermore, many experimental distance CAL systems are established in these information systems. Consequently, the enhanced architecture can improve existing distance CAL systems by the ability to construct a collaborative learning activity. The improvements originate from the server with the following features:

1. *Identify the role of a client:* A client (user) can play many roles, and a set of clients (users) may only play one role. Identifying the role of a client is so important that the server must obtain capabilities and constraints associated with the role from the active database.
2. *Endow capabilities of a role:* The capability of a client is defined according to its role in an activity. For instance, the student's capability specifies that he/she can only propose a position in his/her own group. These capabilities originate from the previous specification of a role in the active database. The capabilities specify all the possible behaviours of a role.

3. *Maintain constraints of a role:* The constraints of a role cover capabilities with a temporal or non-temporal boundary for any particular role. Only when this set of constraints is satisfied can the client (user) undertake a given role.
4. *Autonomous notification to a role:* Notifying a role of some situations is necessary for the server. The notification is made active by the specification of actions in the active database. Autonomous notification is useful to maintain a long-duration activity because there is always someone who forgets to do something on schedule.

#### *Mapping specification to ECA rule of active database*

The collaborative homework example shows how to enhance the client-server architecture with roles and activity. Furthermore, that informal representation can be translated into the following formal syntax in an active database. A situation constraint, called an *ECA rule* in the literature (Widom & Finkelstein, 1990), is defined as a sentence of the form:

```

define rule rule-name
  on <event>
  if <condition>
  then <action>

```

In these <...> items, *First-Order Temporal Logic* (Abadi & Manna, 1990) is applied to handle the temporal issues and specify the temporal constraints of an activity.

FOTL is an extension of first-order logic for temporal inference which has been introduced in the areas of information system design and specification. Because the logical system used in the above example is linear and discrete with respect to the model of time, only two future temporal operators of FOTL are required:

*N<sub>exttime</sub> A* means 'A is true in the next state'; and  
*A U<sub>ntil</sub> B* means 'A is true until B is true';

where *A* and *B* are formulas. Other temporal operators, such as *Always*, *Eventually*, can be expressed with the basic operators.

Now in the collaborative homework example, the first two rules for active notification must query the transcript of the IBIS discussion activity with a temporal issue. If the query results satisfy the expected conditions, the system will detect the temporal condition. When the temporal condition is satisfied, the action is triggered. To illustrate how to translate the informal representation into a formal syntax in an active database, the first active notification can be represented as the following ECA rules:

```
IBIS-Data(Node, Node_type, Date, Link, Link_type, ...)  
define rule AutoNotifyTeacher  
on (select IBIS_Data.Node  
  from IBIS_Data  
  where Node_type = ISSUE and  
    Link_type = NULL)  
if not(true Until (IBIS_Data.Date - Date > 2))  
then notify teacher
```

### Conclusion

This paper has:

- analysed the requirements to construct collaborative learning activities for distance CAL systems;
- depicted why the active database can satisfy the requirements by a collaborative homework example, and
- proposed the procedures for using an active database to construct collaborative learning activity in distance CAL systems.

The results indicate the requirements to construct collaborative learning activity including constructing activity, automating agenda, and querying transcripts of a learning activity. The active database has distinctive advantages and additional facilities in contrast to other techniques for developing distance CAL systems. Additional facilities have been applied, i.e. rule base, event monitor, and execution model, to the collaborative homework example and obtained distinctive advantages, i.e. event-based interaction, defining regulation as rules, and temporal trigger. Furthermore, the required software architecture and procedures for implementation have been discussed. The procedures demonstrate how active database technology can be used to construct an activity and automatically execute the agenda of a collaborative learning activity in the required software architectures. Finally, there is an attempt to map the specification onto the notions used in the discipline of active database development to expound the feasibility of the methodology.

Most importantly, a perspective is provided that will benefit both CAL and Database communities. For the CAL communities, the proposed methodology for a distance CAL system is based on considering the available database technology and requirements of distance CAL systems. For the Database communities, new requirements can be realised, including the required functions, relationships between the functions, and interactions with users, when using a database to support distance CAL systems. In the long-run, it is suggested that a synthesis of the disparate technologies will contribute to the construction of any distance CAL system.

## References

- Abadi, M. & Manna, Z. (1990) Non causal deduction in first-order temporal logic. *Journal of the ACM*, **37**, 2, 279-317.
- Berners-Lee, T., Cailliau, R., Luotonen, A., Nielsen, H.F. & Secret, A. (1994) The World-Wide Web. *Communication of the ACM*, **37**, 8, 76-82.
- Bricker, L.J., Tanimoto, S.L., Rothenberg, A.I., Hutama, D.C. & Wong, T.H. (1995) Multiplayer Activities that Develop Mathematical Coordination. In *Proceedings of CSCL '95 The First International Conference on Computer Support for Collaborative Learning* (eds. J.L. Schase & E.L. Cunnius) pp. 32-39.
- Brownston, L., Farrell, R., Kant, E. & Martin, N. (1985) *Programming Expert Systems in OPS5: An Introduction to Rule-Based Programming*. Addison-Wesley, Reading MA.
- Conklin, J. & Begeman, M.L. (1989) gIBIS: A hypertext tool for exploratory policy discussion. *ACM Transactions on Office Information Systems*, **6**, 4, 303-331.
- Davie, L.E. (1987) Facilitation of adult learning through computer conferencing. In *Proceedings of the Second Guelph Symposium on Computer Conferencing*. pp. 11-22. University of Guelph, Guelph, Ontario.
- Hiltz, S.R. & Turoff, M. (1985) Structuring computer-mediated communication system to avoid information overload. *Communication of the ACM*, **28**, 7, 680-689.
- Hiltz, S.R. (1994) *The virtual classroom: Learning without limits via computer network*. Ablex Publishing Corporation, Norwood NJ.
- Ibrahim, B. (1994) World-wide algorithm animation. In *Proceedings of the First International on the World Wide Web, Paper No 32*. CERN, Geneva, Switzerland.
- Kim, W. (1995) *Modern Database Systems: the object model, interoperability, and beyond*. ACM Press and Addison-Wesley, New York.
- Kunz, W. & Rittel, H. (1970) Issue as elements of information systems. *Working Paper 131, Institute of Urban and Regional Development*. University of California, Berkeley CA.
- McCarthy, D.R. & Dayal, U. (1989) The Architecture of an Active Database Management System. In *Proceedings of the ACM-SIGMOD International Conference on Management of Data* (eds. J. Clifford, B. Lindsay & D. Maier) pp. 215-246.
- Monaghan, J.M. & Clement, J. (1995) Use of Collaborative Computer Simulation Activities to Facilitate Relative Motion Learning. In *Proceedings of CSCL '95 The First International Conference on Computer Support for Collaborative Learning* (eds. J.L. Schase & E.L. Cunnius) pp. 242-246.
- Montague, W. & Knirk, F. (1993) What works in adult instruction: the management, design and delivery of instruction. *International Journal of Educational Research*, **19**, 327-433.
- Sistla, A.P. & Wolfson, O. (1995) Temporal Triggers in Active Databases. *IEEE Transactions on Knowledge and Data Engineering*, **7**, 3.
- Wells, R. (1992) *Computer-mediated Communication for Distance Education: An International Review of Design, Teaching and Institutional Issues*. American Centre for the Study of Distance Education, Pennsylvania State University, University Park, PA.
- Widom, J. & Finkelstein, S. (1990) Set-Oriented Production Rules in Relational Database Systems. In *Proceedings of the ACM-SIGMOD International Conference on Management of Data*, (eds. J. Clifford, B. Lindsay & D. Maier) pp. 259-270.
- Wray, R.E., Chong, R., Phillips, J., Rogers, S., Walsh, W. & Laird, J. (1994) Organizing Information in Mosaic: A Classroom Experiment. *Electronic Proceedings of the Second International WWW Conference '94: Mosaic and the Web, Chicago*.