CHAPTER 3
Idea Storming Cube: an online group creativity support tool

As we mentioned in literature reviews, the challenges for designing a creativity support system include (1) how to build such a system includes different perspectives from domain, field, and individuals, (2) how to stimulate users in generating more paradigm-modifying ideas with appropriate communicative means, (3) how to help users reduce information noise and provide assistance in an iterative and incremental manner to arouse new ideas.

The objective of this research is to 1) build an exciting game-type visual environment for collaboratively creative thinking, 2) effectively stimulate users to enhance the perspective-modifying ability, and 3) build an incremental mechanism to support creative thinking. The goal of the proposed system is to support innovative creativity production, and we will design an experiment to evaluate the learning effect by using this system.

The system that we propose in this study is called *Idea Storming Cube*, which is an Web-based collaborative learning tool in the form of a game for Paradigm-Modifying learning. We will describe the core concept of this system in support of creativity learning and the method which we adopt to reach our goal in this chapter.

3.1 Putting all into a Magic Cube

The associative basis of the creative process is to combine the usual, unusual, and original ideas for incubating innovations [27]; hence, we suggest that the creativity support tool should have a good way to make people feel free to combine any uncritical ideas from him/her, friends, or domain experts. What we want to encourage students to connect inside or outside ideas are as many as the possible combinations of a Magic Cube (Rubik’s Cube) which is about 4.3×10^{19} different permutations. Based on this thinking, we propose to design a computer-based support tool similar to a Magic Cube for incubating new ideas.

Therefore, we need to translate the way of producing innovative ideas into an executable mechanism in the computer world. By this way, we suppose that the knowledge or concept can be translated into an abstract object where people may have different perspectives of one thing, just like the different sides of a square as illustrated in Figure 3-1 (a). In the innovative process, the shift in perspective plays a very important role. So if the system allows us to divide concepts into several pieces of ideas and let people have an effective way to exchange ideas, like in a Magic Cube as shown in Figure 3-1 (b), it will be a good creative support tool for creative thinking. Within the perspective shifting process, we can not only exchange users' ideas, but also involve the knowledgeable concern from domain experts for validating their creative thinking. By combining those different perspectives, it fits in with Csikszentmihalyi's model of creativity. Furthermore, for effective creativity production, we design a suitable circumstance for stimulating perspective-shifting thinking, as to be described in the next section.

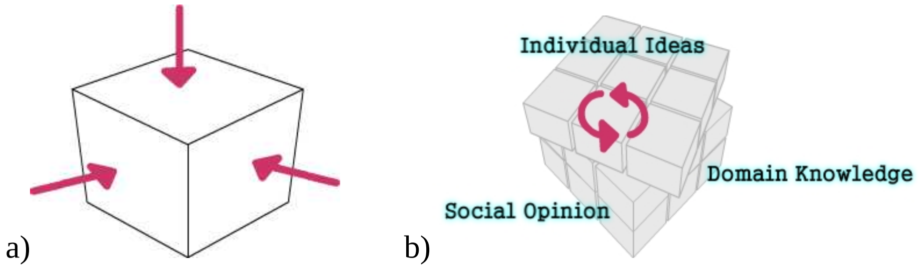


Figure 3-1: a) Different views of one thing

b) Exchanging perpsectives from individual, others, and domain experts' knowledge

3.2 Design of limited view to explore ideas in the brainstorming process

Adopting the Magic Cube as our brainstorming metaphor is to have each user view different subsets of the problem such that they can relate issues in the current context to create new idea. Thus, in our design, users will not be given all information in the brainstorming activity. The main concern of the design is to properly present information as appropriate cognitive stimulation for the users and to avoid providing overwhelming information contradicting people's bounded rationality [36]. As shown in Figure 3-2, each user can always view or write ideas within 9 text fields, just like one side of a 3x3 Magic Cube.

Write idea here		
	Write idea here	
		Write idea here

Figure 3-2: One side of a Magic Cube as seen from a user's view

Only in certain situations, we permit the user to rotate a column or row of the cube for acquiring new information. The rotation mechanism is similar to how the Magic Cube works. For example, User B rotates the first row of the cube to the right, where User C is located, then User A, User C, and User D will give their first rows to User B, User D, and User A, respectively, as illustrated in Figure 3-3.

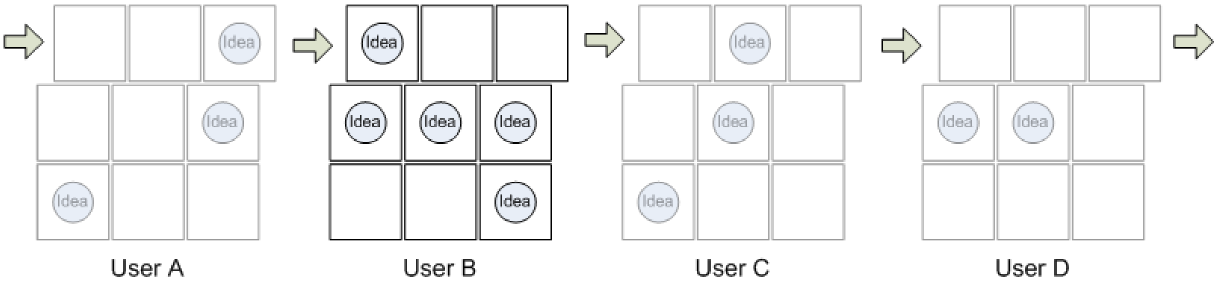


Figure 3-3: Rotating the first row of the cube

However, which are the situations that we should we permit users to rotate in order to exchange the most suitable ideas for them? We have designed a game-based environment with several rules for operation in order to make the process more effective. This game design will be described in next section.

3.3 Design of game-based creativity learning environment

We have designed a playing procedure [21] for this creativity support tool such that users can follow these steps to play with his teammates. The tool works like an online poker game. A particular topic is selected in a game for brainstorming. Each game will require 2 to 6 participants to play. The major steps of the process are described in Figure 3-4.

1) Before the game starts, each one uploads a *Topic Writing file*, a text file for the given discussion topic.
2) Once online, the users will be instructed to write down more ideas.
3) Every user will get a 3x3 array, which is made of 9 cards. Some of the cards may have been filled with ideas (not necessarily generated by themselves), and some may be blank.
4) The system will ask the users to generate ideas from different perspectives and then fill them into the blank cards. The user who proposes the good ideas will get credits after the system validation.
5) For getting more credits, a user can exchange his/her card with others' to get cards of new ideas if you can fill in a blank card with a good idea.
6) An interesting rule is that even if you write an unaccepted idea at present, you may get credits later if someone else has the same idea as you. This encourages you to propose more original ideas, no matter it is right or wrong, as early as possible.
7) Finally, a user wins if he/she has generated the largest amount of valid cards in a given time.

Figure 3-4: Rules for operating the game

Why do we need these rules in the creativity support tool? As Shneiderman and Bostorm pointed out, any creative activity needs well-prepared participants. Thus, we asked the users to upload the *Topic Writing files* (in text format) they have written. Hence, the system can build a draft of the knowledge model for each individual. Based on these models, the system knows which concepts one knows or not. At an appropriate time, the system may recommend a user to acquire other people's ideas if it finds that such differences exist and it is helpful for him. In such a game, every player will strive to generate innovative ideas since it is the way to acquire new idea cards from other players.

In order to let users try to have more Out-of-Box thinking, the system has a special scoring rules as described in Figure 3-5 to encourage users to propose more innovative ideas as early as possible.

R1:A user can get a full credit and a chance to rotate the cube when he/she proposes a new idea which no one has written before and is validated by the system.

R1-1: The suggestion for rotation provides the user with other users' valid ideas that he/she has not seen before.

R2:A user can get a partial credit when he/she see others' ideas that he/she has not seen before.

R3:A user does not get any credit if he/she proposes an idea that has been proposed before.

R4:A user does not get any credit if he/she has proposed an invalid idea in Domain fUP. However, he may get a credit in the future if another user proposes the same idea after him/her.

Figure 3-5: Scoring rules for the game

The R1 rule is to encourage users to write ideas as soon as possible for getting higher credit. The R2 rule is to encourage users to explore more ideas even though he/she may have lost the chances for getting a higher credit because others have proposed first. The R3 rule is to reduce the duplicated ideas. The R4 rule is to encourage users to try more possible thinking potential ideas even it is not valid now.

In contrast to the common communicate- or information-support tools for creative support, by the use of the special playing rules, we try to enhance the effectiveness of such a tool including the frequency of idea generation, the accurate idea complementarities from each person, and the automatic evaluation for incoming new concepts. Hence, in our design, we will use 1) user modeling techniques to build the profile of each user and domain knowledge, in order to record the user's perspectives 2) an artificial intelligent technique to find the most suitable ideas for each one 3) an incremental mechanism of machine learning to evaluate novelty evaluation.

For these goals, we will use a formal user profile, named *formal User Profile (fUP)*, to describe the ideas of the user and other computer agents such as *fUP Analysis Agent*, *User Profile Contrast Agent (UPCA)* to handle the game process. The system architecture will be described further in later sections.

3.4 System architecture

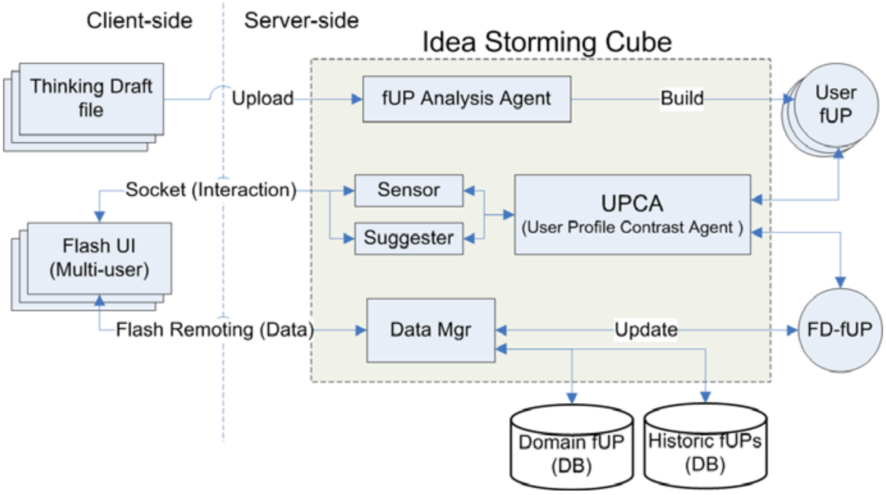


Figure 3-6: System architecture

We adopt a client-server web architecture as illustrated in Figure 3-6. At the client side, we use a Flash user interface to provide rich play interaction, and two data transferring channels: Flash Remoting[1] and ElectroServer[2] Socket protocols. As Figure 3-6 shows, Flash Re-

[1] Adobe Flash Remoting technology is a connection between Flash and web application server to create Rich Internet Application (RIA). http://www.adobe.com/products/flashremoting/

[2] ElectroServer is a socket server that support multi-user with Flash content. http://www.electrotank.com/electroserver/

moting transfers the text data that the users write, and ElectroServer is in charge of user interaction (e.g. rotating the cube) with others.

At the server side, the system provides: 1) an fUP Analysis Agent to analyze player inputs, and to build a fUP for each one, 2) the Data Manager to transfer the data between the users and the DataBase, record the status of the current playing session and continuously update the FD_fUP (i.e., a special kind of fUP to be explained later), 3) the UPCA to compare with users' fUP and FD_fUP to get the discrepancy among users. The design of these agents will be described in the next section.

3.5 User modeling in ISC

3.5.1 Formal User Profile (fUP)

Based on Csikszentmihalyi's model of creativity, we will explicitly model the domain and the individuals such that we can compute the differences among the individuals in order to provide accurate support in a social manner (i.e., the *Field* perspective in Csikszentmihalyi's model).

We use a simplified model of *formal User Profile (fUP)*, proposed in our previous work named User Problem-Solving Ability Modeler (UPSAM) [43], to model domain knowledge for a particular task of idea generation. By the use of Information Retrieval (IR) technologies to extract main ideas from a user's open-end text input, an automatic scoring agent of UPSAM can match a user's responses against the experts' standard answers to computer their similarity [44]. By representing experts' and users' thoughts in problem solving, including both ideas and reasons in the formalism of bipartite graph, we model ideation and explanation as two disjoint sets connected with undirected edges as shown in Figure 3-7.

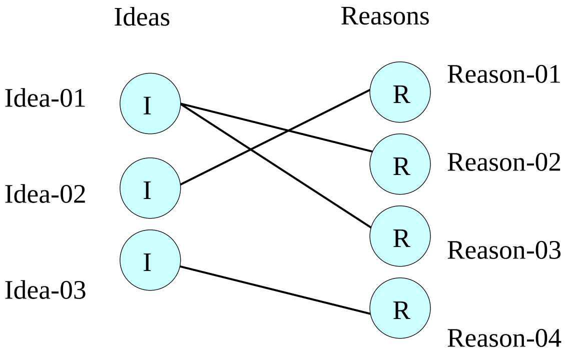


Figure 3-7: Bipartite-graph-based user models of Idea-Reason pair

Each simplified fUP has two finite sets: fUP = {{I_1, I_2,…, I_m}, {R_1, R_2,…, R_m}}, in which I_i represents a specific idea and R_i represents a specific reason in support of an idea. The *Domain fUP* is a special fUP developed by a panel of domain experts to capture ideal solutions for the given task as shown in Figure 3-8. We collect and organize 19 concepts as unique ideas of the Domain fUP and classify them into 5 categories: geology, ecology, natural factor, artificial development and policy. It serves as a fundamental component for the system to support the mapping between an individual and the Domain fUP.

```
<upsam>
   <part id="part1" desc="what are the factors enabling a debris flow to occur?">
      <pairs>
         <pair idea_idref="p1-idea01" reason_idref="p1-reason01"/>
         <pair idea_idref="p1-idea02" reason_idref="p1-reason01"/>
      </pairs>
      <ideas>
         <idea id="p1-idea01" score="4">the location is on a steep dip slope</idea>
         <idea id="p1-idea02" score="2">a rainstorm</idea>
      </ideas>
      <reasons>
           <reason id="p1-reason01" score="4">the friction force between rocks and the
           earth surface is reduced, such that a debris flow is possible to occur</reason>
      </reasons>
   </part>
</upsam>
```

Figure 3-8: A partial Domain Model authored in XML [44]

We have conducted an experiment for testing the automatic scoring ability of UPSAM in a public high school in middle Taiwan, with 70 students involved in the evaluation [20]. The result derived in is study by comparing the results by automated scorers and two human graders reveals that the Pearson’s correlation *r* achieves .87 and .92, which indicates a high and positive correlation. An important feature of the UPSAM is that it maintains an *expert model* representing the domain experts’ knowledge and vocabulary against the problem-solving tasks, and it serves as a useful module in processing users’ open-ended answers for automated grading.

Therefore, UPSAM allows us to compare two bipartite-graph-based user models to address research questions such as *how well are this student’s answers by using the expert’s answers as the criteria* (i.e., essay grading) and *what concepts are known/unknown by the student* (i.e., learning diagnosis).

In ISC, we extend the model of *fUP* as an accumulative model called *Field-Domain fUP* (*FD_fUP*) which is a conceptual space integrated from the collection of individual fUPs and Domain fUP. The objective of *FD_fUP* is to record all ideas from each user’s input history and find the new ideas by comparing the accumulation of historical ideas. It is built from domain expert knowledge and with the society perspective described in Csikszentmihalyi's model of creativity. In other words, the *FD_fUP* is a set of ideas capturing the collective knowledge of the domain and can be described as:

$$FD\_\ fUP = fUP_d \cup fUP_1 \cup ... \cup fUP_n$$

The fUP_d is the domain model and $fUP_i, i = 1...n$ denotes the profile of each individual. The $FD\_\ fUP$ can be reused for the subsequent runs, and it can be continually updated from the results of pervious brainstorming. Thus, we convert people’s thoughts into a collection of idea nodes, and can use them to support the users to generate new ideas.

We not only reuse the fUP model for problem-solving problems, but also adopt the automatic scoring module in our creativity support tool as the fUP Analysis Agent and User Profile Contrast Agent (UPCA). These two agents will be described in details in the next two sections.

3.5.2 fUP Analysis Agent

The main function of the fUP Analysis Agent is to extract ideas from the text files provided by the user. The fUP Analysis Agent works like the automatic scoring agent in UPSAM. It not only grades the users' scores, but also records the ideas of each user. In the ISC game, Individual fUP are built by the fUP Analysis Agent from the Topic Writing file uploaded by the users as shown in Figure 3-9, and it can provide a basic sketch of each individual for which a support decision can be made.

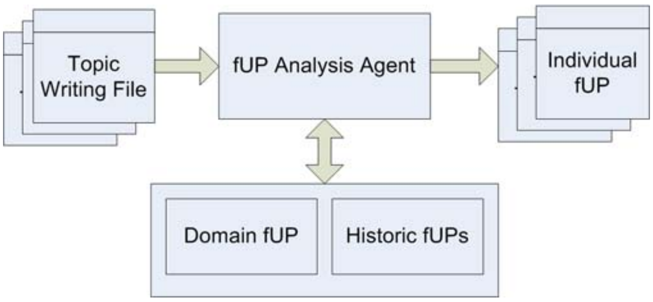


Figure 3-9: Building a preliminary user profile from the text file that the user uploads

The procedures for building individual fUP are revised from UPSAM [44] in order to reach the goal of being interactive. They are described in details below.

```
Procedure: Idea_Identification
Input:   $F_A$: a natural language text file from User A
         $T$: a collection of synonyms in the domain
Output:  $fUP_A$': User A's updated fUP
    $I$ = Segmenting_Processing($F_A$, $d$) //segmenting F into sentence sets I by dilemma d
    for each $i_i \in I$ do
        $i_i' \leftarrow$ Wording_Processing($i_i$, $T$) //remove stop words and refine wordings
        $x \leftarrow$ Node_Mapping($i_i'$)
```

```
5.        $fUP_A$’ ← Node_Adding( $fUP_A$, $x$)
6.    end for
7.    return $fUP_A$’
```

Figure 3-10: The procedure *Idea_Identification* for building the individual fUP

Figure 3-10 shows the procedure for building the individual fUP from a text file. The main idea of fUP Analysis Agent is to adopt token-based words vector of Information Retrieval (IR) technology to find the most similar nodes in the domain fUP. In line 1, the Segmenting_Processing() function uses the full stop to separates the text article into several sentences. From line 2 to line 6, it is an iteration that inserts a node into the Domain fUP in each iteration. In line 3, Wording_Processing() remove stop words or meaningless words by using a collection of synonyms *T* for the given task. Then, a clear collection i_i' of the key words from the sentence can be generated. From line 4 to line 5, the Node_Mapping() function compares the collection i_i' with the Domain fUP for finding the most suitable node, and the Node_Adding() function is used to add idea nodes to the user’s fUP. It will check the duplication of ideas within the user’s fUP before the insertion.

```
Procedure: Node_Mapping
Input:   $I$: a natural language entry
         $v$: a subset of vertices, i.e. ideation subset, in the Domain fUP
Output:  $x$: an idea node
1.    tr:= 0.5                      //setting the filtering threshold, $tr$
2.    for each $c_i \in v$ do
3.        $N \leftarrow Desc(c_i)$              //retrieve the text of ideation
4.        $s_i \leftarrow TFIDF(I, N)$          //compute similarity by using TFIDF method
5.        $sim[c_i] \leftarrow s_i$             //associate the similarity score with $c$
6.    end for
```

```
7.  if $\max_{c_i \in V} sim[c_i] > tr$ then
8.      $x \leftarrow \arg\max_{c_i \in V} sim[c_i]$      //select c with the highest similarity to $I$
9.  else
10.     $x \leftarrow$ New_Node($I$)      //build a new node if no $c$ meets the threshold
11. Return $x$
```

Figure 3-11: The procedure *Node_Mapping* for finding the node in Domain fUP

Figure 3-11 shows the procedure for building an idea node from a natural language entry. From line 2 to line 6, an iterative word-matching task of two strings is conducted. In line 3, we acquire each idea description *N* from the Domain fUP. In line 4, we use the token-based distance function *TFIDF* [7] to compare the idea text between the user and the Domain fUP. The TFIDF similarity (i.e., a variant of cosine similarity in IR) is defined as:

$$TFIDF\ (I, N) = \sum_{w \in I \cap N} V(w, I) \cdot V(w, N),$$

where $V(w, S)$ is the normalized weight of each term w in the string S based on the TFIDF term-weighting scheme [7]. In line 5, we put the result of similarity computation into the *sim[c]* array. Then, from line 7 to line 11, the highest score of the similarity will be selected and the corresponding node with the similarity will be returned. If the similarity is not high enough to satisfy a default threshold *tr*, a new node with the text will be created and returned.

```
Procedure: Node_Adding
Input:  $fUP_A$: User$_A$'s fUP
        $x$: a node from the Domain fUP or a new node with the text from the user
Output: $fUP_A$': User$_A$'s updated fUP
1.  $b$ = false                       //$b$ is a flag for duplication judgment
2.  for each $n_i \in fUP_A$ do
```

```
        $iID \leftarrow GetIdeaID(n_i)$    //get the serial number of each idea node
        $b \leftarrow Duplicate\_Check(x, iID)\ ||\ b$    //check whether duplication exist or not
end for
if $b$ is false then
    $fUP_A$’ ← AddNodeAndScore($x$, $fUP_A$)    //add node to $fUP_A$
else
    $fUP_A$’ = $fUP_A$
Return $fUP_A$’
```

Figure 3-12: The procedure *Node_Adding* for adding a node to user’s fUP

Figure 3-12 shows the procedure for adding a node to the individual fUP. It is not difficult to implement the node adding function for the fUP, but the Duplicate_Check function deserves further explanation. Because each node has its unique serial number *iID* in our design, we can examine whether the node with the same *iID* exists in this user’s fUP. If the node exists, the original fUP will be returned instead.

The fUP Analysis Agent with the automatic scoring ability is used to generate the individual fUP for the system. Based on this agent, we can compare different thinking among people. We will introduce the UPCA in next section to explain how it performs the task.

3.5.3 Brainstorming with User Profile Contrast Agent (UPCA)

In order to stimulate users to generate more divergent ideas, the system is supposed to give users different concepts taken from the Field-Domain Model (*FD_fUP*). Therefore, how to manage the ideas of each user in order to exchange their aspects of thinking is a key issue in the brainstorming process. We design the User Profile Contrast Agent (UPCA) for this task.

UPCA can compare all fUP’s to derive the differential sets of ideas among brainstorming peers. In other words, the common ideas and the unique ideas among the users can be identified. For example, if we want to know what part of the ideas owned by User B is what User A does not know, we can compare *fUP_A* and *fUP_B*, which are fUPs for User A and User B, respectively, with the following formula:

$$fUP_{I_b} = (FD\_fUP - fUP_A) \cap fUP_B,$$

By exposing User A to some of the thoughts and perspectives from User B, we think UPCA can create appropriate cognitive stimulations [25] in the brainstorming activities of User A (i.e., UPCA can exchanges their unique ideas for stimulating each other). With the UPCA built in our system, it monitors the idea generation behavior of the users and then attempts to stimulate creative thinking by using it in the game. The game rules as mentioned in section 3.3 are designed for this propose. We will describe how to implement these rules by UPCA below.

- *R1: A user can get a full credit and a chance to rotate the cube* *<u>when he/she proposes a new idea which no one has written before and is validated by the system</u>.*

For the underline part of *R1* rule, we use the following formula:

$$u = fUP_d - (fUP_1 \cup ... \cup fUP_n),$$

where fUP_n, i =1…n denotes the profile of each individual, fUP_d denotes Domain fUP, and u denotes a set including all users' unknown ideas. By this formula, we can verify whether the idea proposed by the user is a new idea or not (i.e., only new for all users here, but new ideas for Domain fUP will be discussed later). We can give this user a full credit and a chance to rotate the cube. But as for how to rotate the cube, we will refer to the *R1-1* rule:

- *R1-1: The suggestion for rotation* *<u>provides the user with other users' valid ideas that he/she has not seen before</u>.*

For the underline part of *R1-1* rule, we use the following formula:

$$s_1 = (fUP_d - fUP_1) \cap (fUP_2 \cup ... \cup fUP_n),$$

where fUP_n, i =2…n denotes the profile of each individual except User$_1$, fUP_d denotes Domain fUP, and s_1 denotes a set of potentially suitable new ideas for User$_1$. From this formula, we know the ideas that can be used as the stimulation for User$_1$. However, to suggest the rotations on User$_1$'s user interface for finding intriguing ideas in other sides of the cube is another issue. We will describe the operation of the ISC in the later sections.

- *R2: A user can get a partial credit when he/she see others' ideas that he/she has not seen before.*

The *R2* rule is to encourage the users to explore as many different ideas as possible. This may result from the rotation by others. The set s_1 in *R1-1* can be reused here to identify the ideas that are unknown to a specific user.

- *R3: A user does not get any credit if he/she proposes an idea that has been proposed before.*

For the *R3* rule, we can use the following formula:

$$e = fUP_d \cap (fUP_1 \cup ... \cup fUP_n),$$

where fUP_n, i =1…n denotes the profile of each individual, fUP_d denotes Domain fUP and e denotes the set of all users' known ideas. By this formula, we can verify whether the idea proposed by the users has been written or explored before or not. If an old idea is proposed again, no credit will be given.

- *R4: A user does not get any credit if he/she has proposed an invalid idea in Domain fUP. However, he may get a credit in the future when another user proposes the same idea after him/her.*

The *R4* rule is an incremental mechanism for encouraging users to suggest new ideas that are not in the Domain fUP. Therefore, how to recognize the similarity between two new idea nodes beyond Domain fUP is an important issue for this purpose. We use the following two procedures to address this issue.

i. Putting the invalid ideas of all user's fUP into the potential idea space

In order to build the potential idea space, we should record the invalid ideas from each user. In fact, the fUP Analysis Agent has done this task by giving each invalid idea a unique idea node with a new serial number (i.e., iID) and the segmented key words from the user's text input. Thus, each user's fUP will record new idea nodes beyond Domain fUP as shown in Figure 3-13.

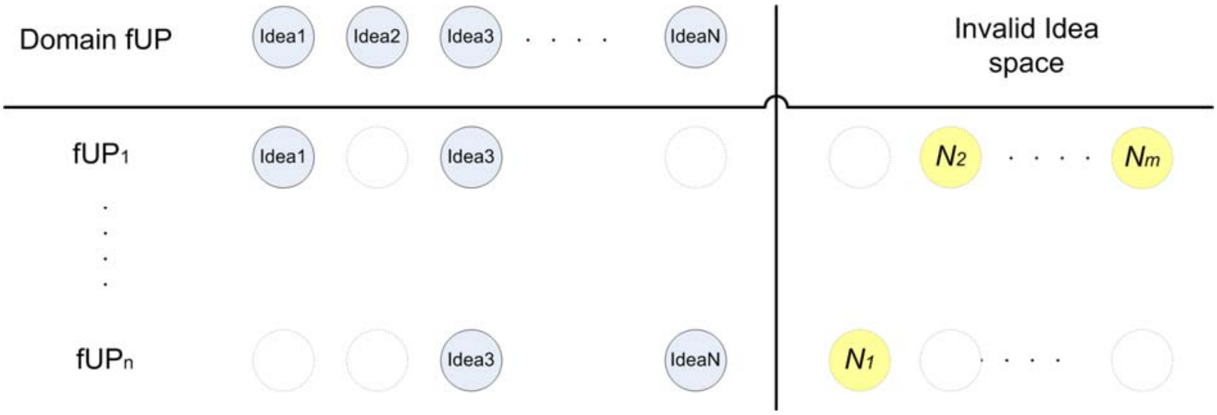


Figure 3-13: The new idea nodes beyond Domain fUP

Then we can use the following formula to compute the set (i.e., the potential idea space s_k, k =1…m denotes different invalid ideas provided by each user) of all users' new ideas that have been invalidated according to the Domain fUP.:

$$s = (fUP_1 \cup ... \cup fUP_n) - fUP_d ,$$

where fUP_n, i =1…n denotes the profile of each individual, fUP_d denotes Domain fUP. By this formula, we can verify whether the new idea proposed by the user can be mapped to the new idea space *s* or not, and the mapping task is described in next procedure.

ii. Computing the similarity between the new idea node and the potential idea space

The mapping of the new idea to the potential idea space is accomplished by the fUP Analysis Agent. The flowchart of the process is illustrated in Figure 3-14.

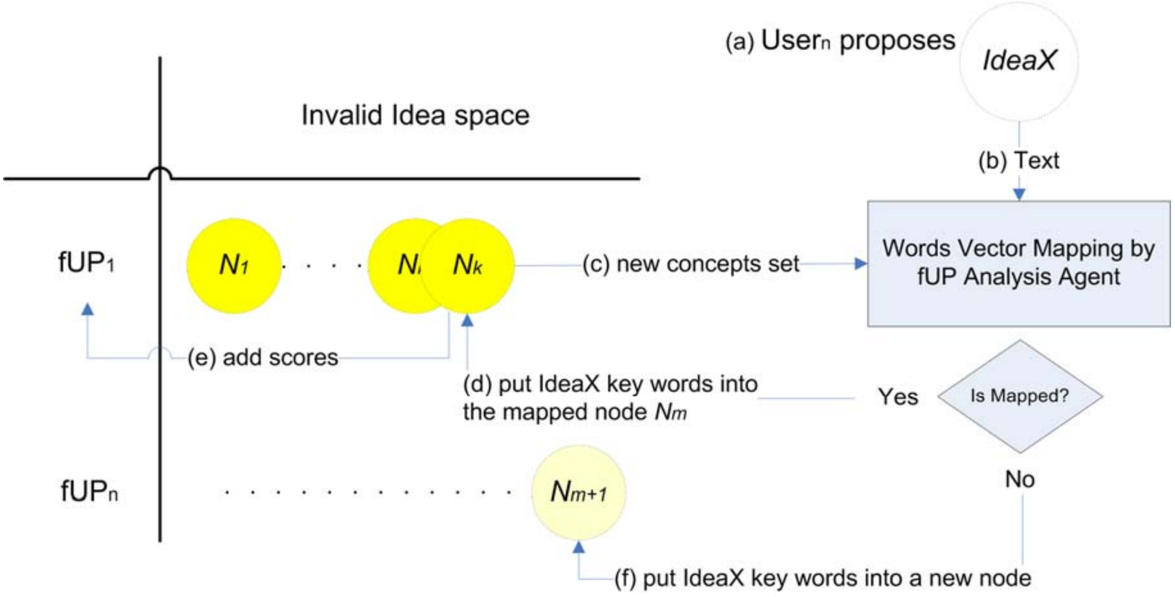

(a) Userₙ proposes
IdeaX
(b) Text
Invalid Idea space
Words Vector Mapping by fUP Analysis Agent
fUP1
N1
Ni
Nk
(c) new concepts set
(e) add scores
(d) put IdeaX key words into the mapped node Nm
Yes
Is Mapped?
No
fUPn
Nm+1
(f) put IdeaX key words into a new node

Figure 3-14: The procedure for mapping a new idea with the potential idea space

In step (a), suppose that if $User_n$ proposes a new idea node *IdeaX*. Then the fUP Analysis Agent can acquire the key words from the text of *IdeaX* in step (b). Then, we can use the Node_Mapping function (i.e. replacing Domain fUP with the potential idea space) of the fUP Analysis Agent to compute the similarity between *IdeaX* and those in the potential idea space in step (c). After computing the similarity between IdeaX and each idea in the set *s*, we can find the idea node N_k with the highest similarity. If the similarity score higher than some threshold *tr,* then the steps (d) and (e) will be executed, and $User_1$ will be accredited. Otherwise, step (f) will be executed and a new node N_{m+1} will to be added into fUP_n.

By the above two procedures, we can verify whether the new idea proposed by a user can become a valid idea. These procedures are used as an incremental mechanism to add new ideas from the current gaming scenario into the FD-fUP, which is employed as the sources for steering adaptive supports in the future. Hence, we have also met the goal of the R4 rule.

In conclusion, the UPCA agent can retrieve, summarize, and synthesize information presented by the current gaming peers and those stored in the historic FD-fUP model.

3.6 Learning strategies: Basic Mode versus Agent-Assisting Mode

We have designed two different strategies: *Basic Mode* (BM) and *Agent-Assisting Mode* (AAM) for the operation modes with and without incorporating a virtual agent to manipulate the brainstorming activities, respectively. The contents that the recommendation system uses for manipulating the cube for idea exchanges will also be affected by the fact that a virtual agent exists or not. The ideas of the two modes and their targeting portions of the ideas in FD_fUP are illustrated in Figure 3-15 and explained in more details later.

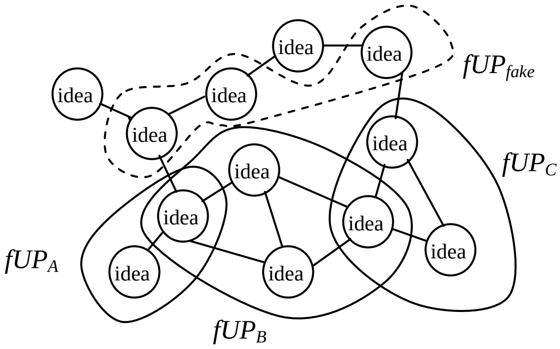


Figure 3-15: Different learning strategies in ISC: BM and AAM

In the basic mode, the system will attempt to lead the users to explore more domain knowledge that he/she does not possess based on the kernel design notion of knowledge sharing. The goal is for the users to share the same good understanding of the domain as her/his peers. Therefore, in each run of the game, the ISC system identifies what a user has learned according to his/her fUP and presents new ideas from his/her teammates such that all team members can learn to understand all the ideas that have been accumulated in the idea storming activities. For example, for user U_A in Figure 3-15, the target portion of FD_fUP falls on the ideas of U_B and U_C that are already owned by U_A. Likewise, when getting a chance to exchange ideas, U_B will be exposed to the idea owned by U_A and U_C. An adaptive feedback mechanism is designed to support this kind of knowledge sharing and achieve group consensus eventually.

In the agent-assisting mode, the system encourages users to produce more perspective-modifying thoughts. In ISC, we use a fake user U_{fake}, a perspective-modifying thinker, to take the real users away from their original thinking box. The fUP of U_{fake} is a subset of FD_fUP that does not intersects with any of the fUP's (fUP$_\mathrm{i}$, i=1,2…n) of the real users, as expressed in the following formula:

$$fUP_{fake} \subseteq FD\_fUP - \bigcup_i fUP_i$$

As depicted in Figure 3-15, fUP$_\mathrm{fake}$ is a dynamic set that is updated during the activity according to the ideas that have been created by the users. The objective is for all users to collaboratively explore all of the ideas in FD_fUP. In addition, each user should explore different parts of the FD_fUP such that the overlap between individual fUPs can be minimized for productivity and efficiency.

3.6.1 Basic mode interaction design

It is worth to explain how we design the interactive mechanism for users to exchange their ideas. In the original game rules of a Magic Cube, the player can arbitrarily rotate the cube in any row or any column. In our game, however, each user only can view one side of the cube and cannot freely choose any row or column to rotate. The searching algorithm for finding good ideas for a user is mentioned in subsection 3.5.3, but these good ideas disperse in different sides of the cube. It is a problem how to bring these ideas into a specific user's view by rotating the cube only once.

Therefore, a heuristic method which is used to solve this problem is that we only consider the fUPs of the user's neighbors. In fact, each user will face one side of the cube, so they have their physical positions respectively. According to the position of the neighbors, we can find the rotation directions for the user to get the good ideas as the yellow arrows show in Figure 3-16 (A).

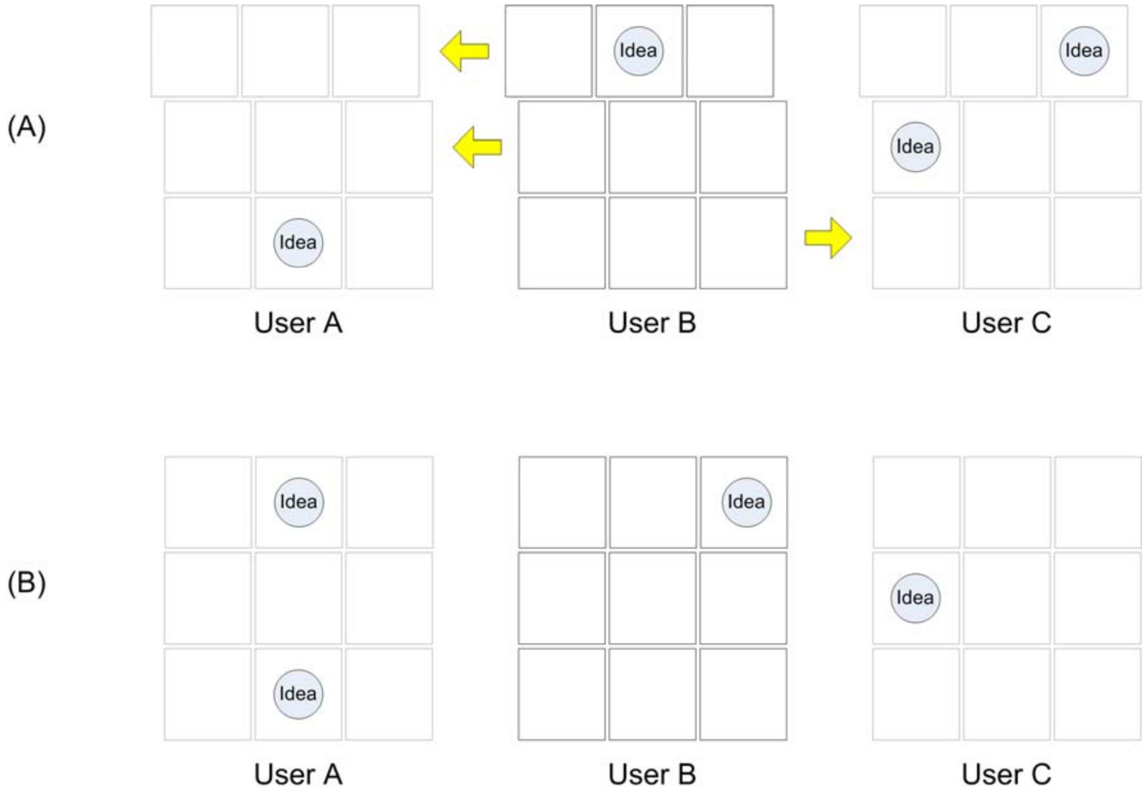


Figure 3-16: Showing yellow arrows before rotating and after rotating the cube

After rotating the cube, the neighbor's good ideas which are suitable for this user will appear in his view as shown Figure 3-16 (B). Besides, all suggestion arrows will disappear after rotation because the chance is only once. Note that the chance can not be extended to the following steps even the user suggests many good ideas.

3.6.2 Agent-assisting mode interaction design

Although we know the supports that an agent can give to a user, how to design the assisting mechanism is another issue. For example, what time should the agent intervene a users' thinking and how much support that an agent should give?

For the Paradigm-Modifying phenomenon, the coverage rate of idea category may be a good measure to tell whether the user is in a thinking plight. We have classified ideas into categories in the Domain fUP, so we can easily calculate the coverage rate from the group or each individual. We use it to assist users by giving an unexplored idea in the category with the lowest coverage rate. Note that we will support both group and individual because each user may lack different concepts and may not get appropriate supports if only the group is considered.

Another import issue here is deciding the right time to give the support. In our system, we use the ratio of valid idea over all ideas to determine whether the user needs supports from the peer-like agent. The system will record the validity status of a user's ideas and immediately calculate the ratio R_v of valid idea over all ideas. Therefore, in our design, when the R_v is lower than some specified threshold, the peer-like agent will start up to give the user appropriate supports, but the supports will be suspended when R_v is higher than the specified threshold.

Procedure: Agent_support

Input: *fUP*: Group's fUP (or individual's fUP for individual support)

Output: *x*: a node from the Domain fUP

```
tr = 1                            //setting the filtering threshold, tr
for each $n_i \in fUP$ do         //$n_i$ is the node from the fUP
    $if\,(isValid(n_i))$          //get the valid status of each idea node
        validCount ++
    else
        invalidCount ++
7.  end for
8.  $R_v$ = validCount / (validCount + invalidCount)
9.  if $R_v < tr$ then
10.     $x$ = SearchIdea( smallCat(fUP), Domain fUP)
11. end if
12. return $x$
```

Figure 3-17: The procedure *Agent_Support* for generating a stimulating idea.

Figure 3-17 shows the procedure for determining the right support time and generating a stimulating idea by inputting the group’s or individual’s fUP. Note that the group’s fUP denotes that the union set of each individual’s fUP in their group. From line 2 to line 7, the number of valid idea in the fUP is computed. In line 8, we calculate the ratio R_v of valid idea over all ideas. In line 9, if the ratio R_v is not higher a threshold *tr* (i.e., the user need support), a node from the Domain fUP will be created and returned in line 10. Note that the function SearchIdea needs to use another function smallCat to find the category with the smallest coverage in the user’s fUP, and then find the appropriate idea in the category of the Domain fUP to return.

A special design here is that the agent support is not only for the group, but also for each individual by inputting each individual’s fUP. In the Aggent_Support algorithm, the input is the fUP format, no matter it is from the group or each individual. For the group support, the peer-like agent within the group will generate a new idea automatically from the Domain fUP with the yellow idea nodes in the agent’s side of the cube as shown in Figure 3-18 (A). Other users rotating the cube will naturally get the idea support from the peer-like agent.

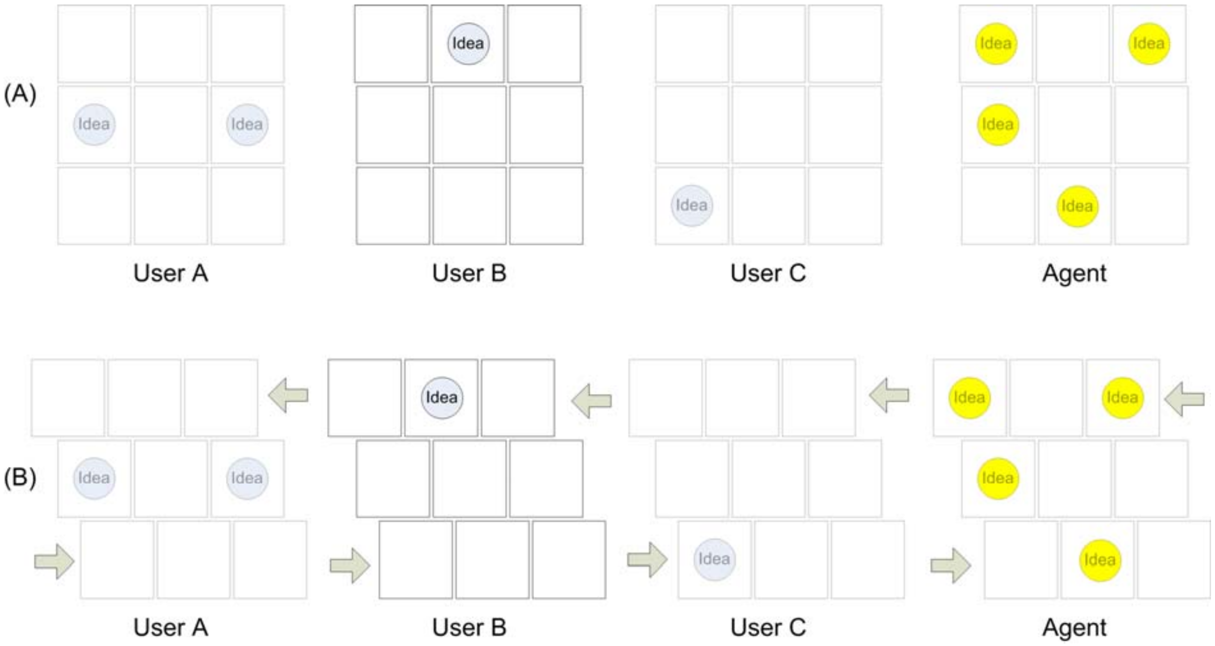


Figure 3-18: (A) Generation ideas in the agent's view (B) obtaining new ideas from the agent after rotating the cube

For the individual support, our system does not obey the physical constraint of the Magic Cube in the game. In the original game design of Magic Cube, when you turn the row four times in the same direction, the row will come back. However, it is difficult to support each individual under this constraint because the most appropriate ideas for him may not exist in his neighbors. Therefore, we have designed a special mechanism for manipulating the cube for this problem. When the agent accepts the support request from a user, the agent will only exchange his row or column with the agent's without affecting other users' as shown in Figure 3-19.

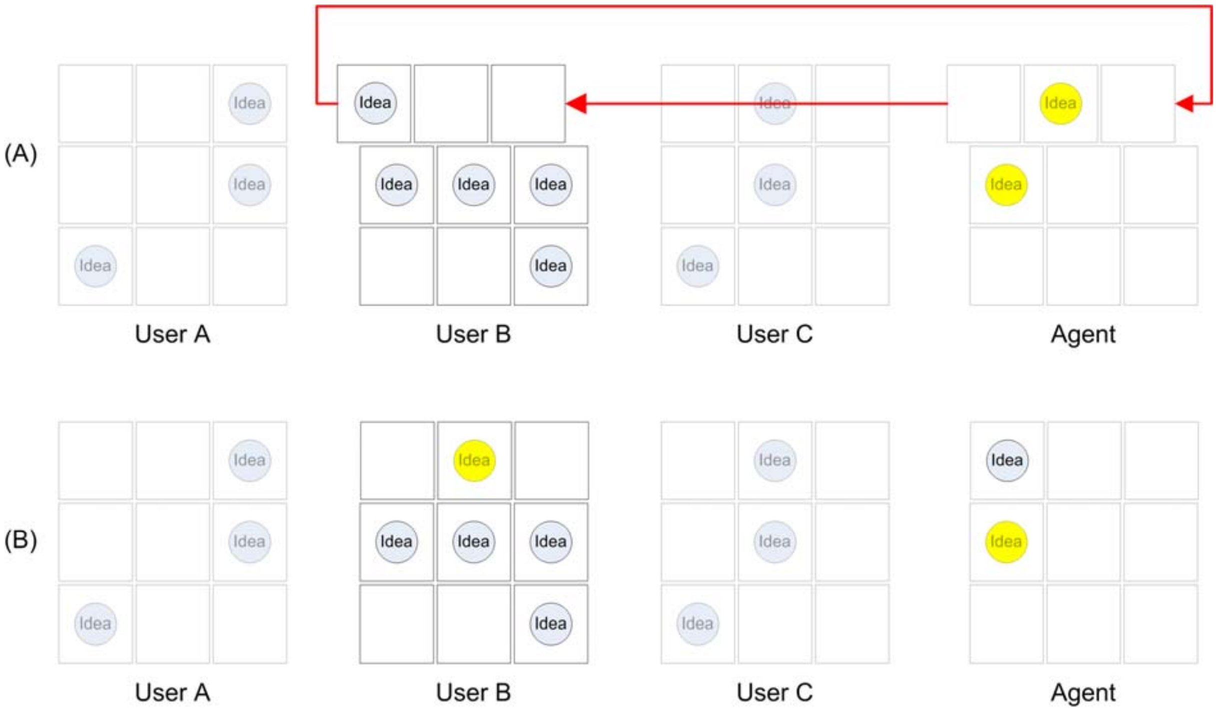
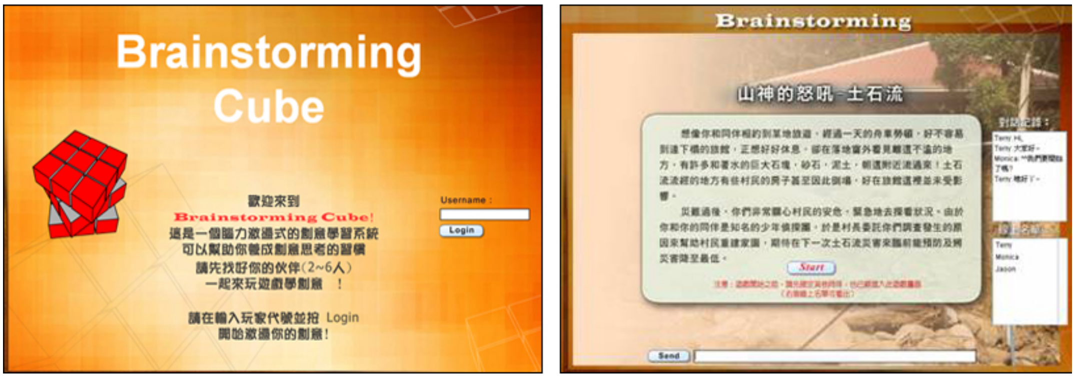


Figure 3-19: (A) Only rotating certain user's view with agent (B) after rotating

3.7 Use Case

Here we use the scenario of learning the conceptual knowledge for the topic of *Debris Flow Hazard* problem in the Basic Mode (BM). The system has been implemented with a client-server architecture such that multiple users can log into the system concurrently for the brainstorming activities, as shown in Figure 3-20.


Brainstorming Cube

歡迎來到
Brainstorming Cube!
這是一個腦力激盪式的創意學習系統
可以幫助你養成創意思考的習慣
請先找好你的伙伴（2~6人）
一起來玩遊戲學創意　！

請在輸入玩家代號並按 Login
開始激盪你的創意！

Username :

Brainstorming

山神的怒吼－土石流

想像你和同伴相約到某地旅遊，經過一天的舟車勞頓，好不容易到達下榻的旅館，正想好好休息，卻在落地窗外看見離這不遠的地方，有許多和著水的巨大石塊，砂石，泥土，朝這附近流過來！土石流流經的地方有些村民的房子甚至因此倒塌，好在旅館這裡並未受影響。

災難過後，你們非常關心村民的安危，緊急地去探看狀況。由於你和你的同伴是知名的少年偵探團，於是村長委託你們調查發生的原因來幫助村民重建家園，期待在下一次土石流災害來臨前能預防及將災害降至最低。

注意：遊戲開始之前，請先確定其他同伴，也已經進入此遊戲畫面
（右側線上名單可看出）

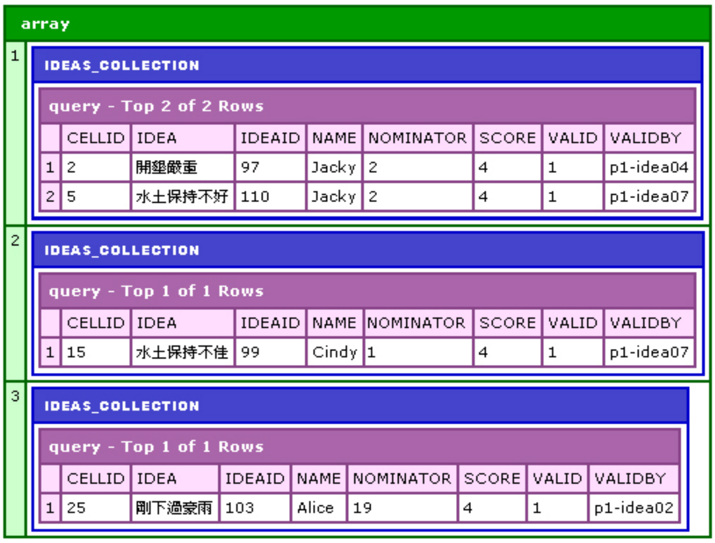
對話記錄：
Terry: Hi,
Terry: 大家好~
Monica: ^^我們要開始了嗎?
Terry: 嗯好了~

線上名單
Terry
Monica
Jason

Figure 3-20: Screenshots of Login and instruction

After the players log on the cube, in our game design, each player first submits his/her ideas via uploading a text file containing the initial ideas about the possible causes of debris flow that each user has before the brainstorming activities. These initial ideas are used to build FD_fUP along with the domain knowledge collected from experts.

Each user is arranged to face one side of the cube. After the game starts, each user is asked to input as many ideas (causes of debris flow) as possible. When a novel idea is recognized, the user is allowed to rotate a row or a column of the cube in order to exchange ideas with others. The winner would be the one who can create or view as many ideas as possible when the game is over after a fixed period of time. In other words, a user can gain credits by creating novel ideas by him/her or by rotating the ideas of other users as the reward of contributing a new idea. Therefore, each player will strive to be productive in idea generation in order to win the competition.



array

1

IDEAS_COLLECTION

query - Top 2 of 2 Rows

	CELLID	IDEA	IDEAID	NAME	NOMINATOR	SCORE	VALID	VALIDBY
1	2	開墾嚴重	97	Jacky	2	4	1	p1-idea04
2	5	水土保持不好	110	Jacky	2	4	1	p1-idea07

2

IDEAS_COLLECTION

query - Top 1 of 1 Rows

	CELLID	IDEA	IDEAID	NAME	NOMINATOR	SCORE	VALID	VALIDBY
1	15	水土保持不佳	99	Cindy	1	4	1	p1-idea07

3

IDEAS_COLLECTION

query - Top 1 of 1 Rows

	CELLID	IDEA	IDEAID	NAME	NOMINATOR	SCORE	VALID	VALIDBY
1	25	剛下過豪雨	103	Alice	19	4	1	p1-idea02

Figure 3-21: The ideas generated by the users as recorded by ISC

The system records the ideas that each user has inputted as shown in Figure 3-21. The UPCA compares the ideas with fUP_d, but also with other users in order to find the ideas that can be used to inspire a specific user. For example, in the example of Figure 3-21, the fUP's of all the users are the following:

fUP_{Jacky} = {idea04, idea07}

fUP_{Cindy} = {idea07}

fUP_{Alice} = {idea02}

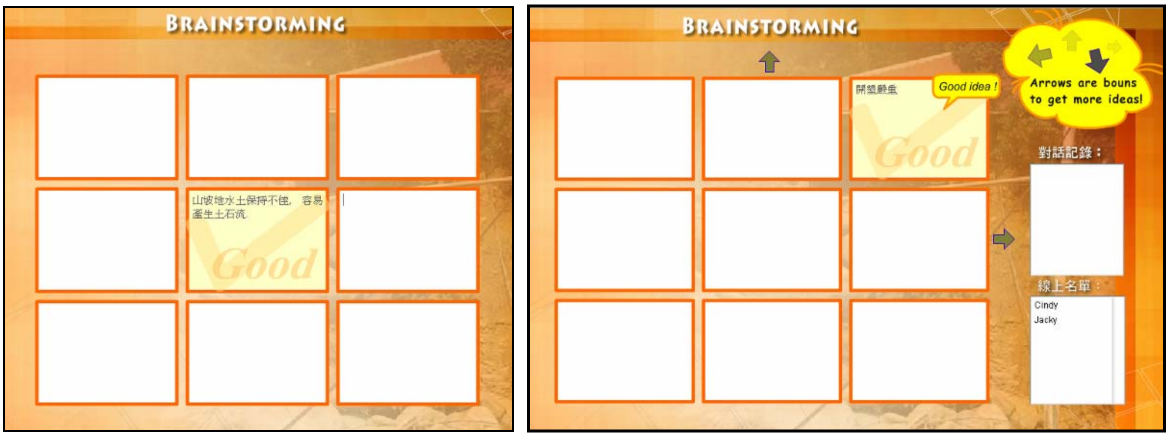
The UPCA uses the following formula to find the new ideas for Cindy:

$$fUP_{IdeaForCin\,dy} = (FD\_fUP - fUP_{Cindy}) \cap fUP_{Jacky} \cap fUP_{Alice}$$

The resulting idea set is {idea04, idea02} with one idea from Jacky and another from Alice. The system can then locate the cards containing these ideas in the cube, and then display blinking arrows to propose rotation suggestions to Cindy for getting these ideas.

Finally Cindy will be allowed to rotate a row or a column of the cube at her will to exchange ideas with others. The winner would be the one who can contribute as many novel ideas as possible when the game is over. In other words, a user can gain credits by contributing novel ideas that have not been proposed before. As a reward, the user is allowed to rotate in a row or column to get a better understanding of information in the cube. Consequently, each player will strive to be productive in idea generation in term of speed and quality in order to win the competition.

The cube will continuously compute each player's fUP to make it as a support to others. However, the system does not rotate the cube voluntarily for the users. Instead, in the competition, the system only prompts the user the possible rotations that he/she can make to attain new ideas, as shown in Figure 3-22. The final decision on what to rotate to acquire more inspiring ideas from others still relies on the users.



山坡地水土保持不佳，容易產生土石流

Figure 3-22: Providing rotation arrow by dynamic computing XOR set of user's fUP

For example, if a user named 'John' has inputted a novel idea, John will be allowed to rotate the cube once with the suggestions arrow. Similarly, the cube will also suggest others to take away this new idea card by John when they get a chance to rotate the cube. In Figure 3-23, we show the interface where the John entered a novel idea in the upper middle cell and the suggestion arrows for exchanging this node will be shown in another user's (Mary) window. As soon as the user input a novel idea, he/she will get a chance to rotate this node into the current view. Therefore, the system encourages the users to input their ideas as quickly as possible such that the contributed ideas can be considered "novel" at that moment.

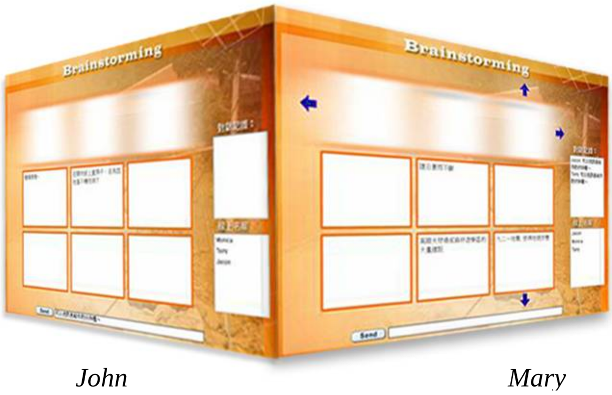


Figure 3-23: A user gets another's idea card by rotating a row

Through the competition, we hope that the user can strive to come up with high-quality novel ideas quickly. By limiting the view of a user to a few ideas, the system allows a user to concentrate on generating related ideas based on the current view. In other words, the system makes use of the concept of rational boundary [36] to increase the quality of generated ideas, and it also uses the cube mechanism to gradually change the view to inspire new ideas.