Using a Game for Social Setting in a Learning Environment: *AlgoArena* — A Tool for Learning Software Design

Hiroshi Kato and Akiko Ide

Information Technology Research Laboratories, NEC Corporation

Abstract

AlgoArena is a tool for education of software design to beginners. The purpose is to develop the ability to think algorithmically and the ability to view things systematically. It is a simulation game of sumo wrestling — the traditional Japanese national sport. Students are supposed to program the actions of their own wrestler using a programming language so as to win matches against other wrestlers. To make the wrestler stronger, students are encouraged to analyze the chaotic situation systematically, devise better tactics, and incorporate them into the program. Such comprehensive problem-solving activities are comparable to the processes involved in software design. In addition, students commit to solving their own problems at their own responsibility. It is also comparable to the authentic activities of a software designer. In this sense, activities with AlgoArena can be regarded as a significant 'epitome' of authentic software design.

Social setting for the learning environment is attributable to constraints embedded in the system of AlgoArena. For example, owing to the constraint of the game that a player needs opponents to fight with, students are encouraged to have matches with others, and social interaction among students is facilitated. In this way, a game situation is brought into the context of learning and produces a community of learners.

Keywords — simulation game, collaborative learning, situated learning, computer literacy, programming language, software education, Logo.

1. Introduction

According to a theory of situated cognition, learning is regarded as a process of enculturation [1], and it occurs as newcomers gradually increase their participation in a community of practice [3]. Learning is thus inseparable from practice. From this viewpoint, Brown pointed out one problem of schooling as follows:

> *Most classroom activity inevitably takes place within the culture of schooling, but it is attributed by both teachers and students to the cultures of readers, writers, mathematicians, historians, and so on. What students do in school thus tends to be a sort of ersatz activity, distorting both what is learned and the culture to which it is attributed [1].*

The most rigid interpretation of this claim would demand 'on the job training (OJT)', i.e., an educational method in which students take part in actual practice at the workplace, such as in conventional apprenticeships. In reality, however, it is not always possible for students to become members of the actual community as apprentices only for the learning experience. First, from the viewpoint of the workplace, the community's capacity to accept excessive numbers of apprentices is inadequate by ordinary. The community of practice is not able to afford to educate students who are not expected to become regular member. Second, from the viewpoint of students, they will have less opportunity to experience various cultures, because OJT is generally a very time-consuming process, whereas they should be enabled to encounter a wide variety of cultures. For these reasons, it would be impracticable to educate everyone on the basis of OJT.

As a result, the authors have left the framework of conventional schooling as it is, and have tried to bring realistic culture of practice into culture of schooling. Although practice at school cannot be the same as it is at the workplace, we believe that it is possible to inject student's activities with 'reality' of actual practice. Before getting into the details of our approach, let us clarify what we mean by 'reality'.

The authors think that actual experiences have at least two aspects. One aspect of reality is perceptual reality. It concerns the similarity between perceptions caused by physical stimuli in the real world. Virtual reality accomplishes reality in this sense. For example, when a student works with the same tools that established members use for authentic activities, perceptual reality for the activity is high in the sense that he/she can feel, touch, hear, smell, and see the same things as the members do.

On the other hand, there is another aspect to actual experience, namely, social reality. It concerns the similarity between the social commitments created by acts. Every act inevitably produces some social commitment to others. For example, articulating "This is a pen." is not only a description of the state of affairs, but produces a speaker's commitment to a listener regarding the existence of the pen. That is why the speaker is anticipated to disprove, if someone says, "It is a pencil." or "I cannot find a pen." Out of the network of such commitments, meanings for acts emerge. Therefore, it can also be stated that social reality concerns the similarity between meanings of acts.

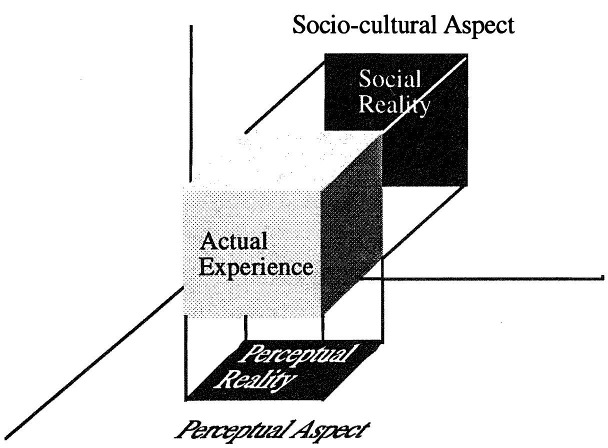

Figure 1. Relationship between social reality and perceptual reality.

Both aspects of reality are, connected by actual experience, orthogonal as shown in Figure 1. That is, there can be situations that have high perceptual reality and low social reality at the same time, or vice versa.

Suppose that you are participating in a role-play dialog with a native speaker to learn a foreign language. You can listen to the fluent pronunciation of the native speaker, look at his/her gestures, and even shake hands. What you perceive there is much the same as what those who speak the dialog as a part of their authentic activities perceive, so it has a high perceptual reality. However, such a speech-act does not produce a similar commitment such as an authentic speech does. For example, articulating "Let's meet there then." in real life usually commits a speaker to the listener such that the speaker must go to the designated place at the designated time. However, if the words are spoken in the role-play dialog, the speaker is not expected to do so. This is why the act in the role-play has low social reality.

However, when you are communicating via a computer terminal using a chat program (a real-time communication system with characters and alphanumerics on screen), such as 'talk' in UNIX, you are unable to listen to live voices, or look at the speaker's facial expressions. Instead, you are forced to read strings of featureless characters. Therefore, the act has relatively low perceptual reality, compared to ordinary conversation. However, sending a string of "Let's meet there then." produces the sender's commitment to a receiver for carrying out the appointment, which is as valid as the appointment by face-to-face talk. Hence, the act has high social reality.

Regarding education, perceptual reality can be supported by various media in a learning environment. In particular, multimedia technology, such as full color images and high fidelity sounds, can contribute primarily to this aspect of reality. Meanwhile, social reality is affected by the social settings of a learning environment. Specifically, it depends on the means of organizing student activities, putting them into context, and maintaining the community of students.

As Brown pointed out in the earlier quotation, activities at school tend to be a somewhat ersatz. This is partly because the environment in which activities take place is quite different from the workplace, i.e., lack of perceptual reality. Another reason that should be taken into account is the lack of social reality. That is, students at school tend to commit to their teacher for learning activities, while established members of the community of practice commit for the achievement of their own tasks at their own responsibility.

Although both aspects of reality are essential for learning, the authors would like to emphasize the importance of social reality rather than perceptual reality. Regarding perceptual reality, recent progress in computer technology has improved the learning media to a considerable extent. However, it still depends on the expertise of experienced educators to produce learning activities that are rich in social reality. Therefore, we are convinced of the necessity to develop a design methodology for the social setting of a learning environment allowing the accomplishment of social reality.

From the reasons mentioned above, the authors have tried to design a social setting using the constraints provided by learning tools, such that students, wishing to establish their identities in the community, will be able to work on practices to achieve their own tasks at their own risk, and not solely to maintain a good relationship with their teacher.

Table 1. List of reserved words in the programming language.

action commands	move_forward move_back bend_forward bend_back step_forward step_back throw slap_down push_forward grasp_mawashi disturb_hishand release_mawashi wait
system variables	my_position his_position distance my_posture his_posture my_arm his_arm upper_arm my_leg his_leg
commands and special symbols	if ifelse select endselect case caseelse repeat while until to end stop stopall make local table tmake search and or not # + - * / > < >= <= =

2. AlgoArena: a tool for learning software design

2.1. Discussions on student's activities

AlgoArena [2] is an introductory tool to teach software design to beginners. The authors focus on developing the algorithmic way of thinking and a systematic view of things through problem-solving activities, rather than just teaching a specific programming language.

AlgoArena is a simulation game of sumo wrestling — the traditional Japanese national sport. Students are supposed to program the actions of their own wrestler using a LOGO-based programming language to win matches against other wrestlers.

In the AlgoArena system, some constraints for the social setting of the learning environment are incorporated, so that students can form their community of practice and facilitate social interaction. For example, ① Using the constraint that a player needs opponents to fight with, students are encouraged to have matches with others, and social interaction among students are facilitated; ② Using the constraint that one wins while the other loses, the player's will to win is evoked, and students come to share a common purpose — to try to win — and a common sense of values; ③ Using systematic match organization, such as tournaments or leagues, students are able to have an exciting time together, fostering the sense of community; and ④ Using a definite game system that requires deep thought, which will be mentioned later, the community is kept lasting and evolutive. In this way, AlgoArena can help to form a student's community.

Regarding activities in the game, students are encouraged to systematically analyze why they lost, to plan tactics to win, and to implement them into the programs, while simultaneously keeping various constraints in balance. These comprehensive problem-solving activities are comparable to authentic activities in software design, i.e., in analyzing situations, defining problems to solve, devising ways to solve them, and implementing them into the program. Furthermore, what is important here is to what and how students commit through those activities. In AlgoArena, students tend to commit to solve their own problems to establish their own identities in the community at their own risk, rather than commit to a teacher for learning. This is also comparable to the authentic activities of programmers, who are socially motivated and are responsible within the context of solving the specific problem to which they are committed. In this sense, activities using AlgoArena have a high social reality, and can thus be regarded as a significant 'epitome' of authentic software design.

This stands in contrast to the conventional way of teaching software design. A typical course consists of instructions and discontinuous exercises, where solving problems is not valuable in itself, but is valuable only for the purpose of education. This is because they are usually problems that have already been solved. In addition, since problems are presented in a well structured and specifically defined form, students do not have to analyze chaotic states of affairs to find out the problem they need to solve. Consequently, students are more inclined to commit to the teacher for 'learning' than undertake problem-solving itself. From this, the authors would say that the conventional way lacks social reality as well as essential portions of authentic cultural practice.

AlgoArena is designed to be the core of the student's community, in which students practice software design. It does not explicitly tutor like a conventional CAI, nor implicitly embed what is to be learned as principles of the virtual world like a conventional microworld. However, it serves the arena where students apply what they have learned. Students learn mainly through social interaction with a teacher and peers, rather than solely through interaction with a computer.

2.2. AlgoArena System Overview

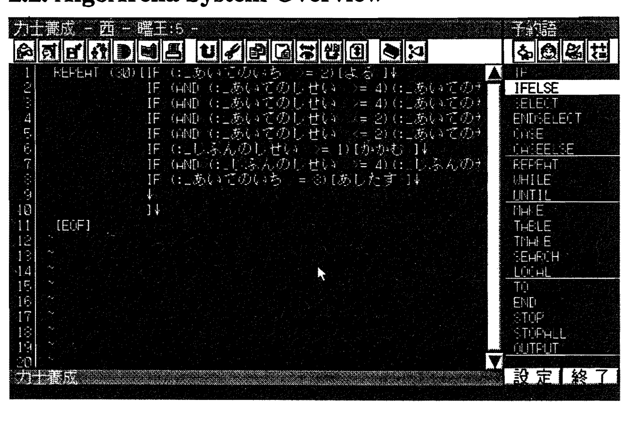

Figure 2. Screen editor incorporated in AlgoArena.

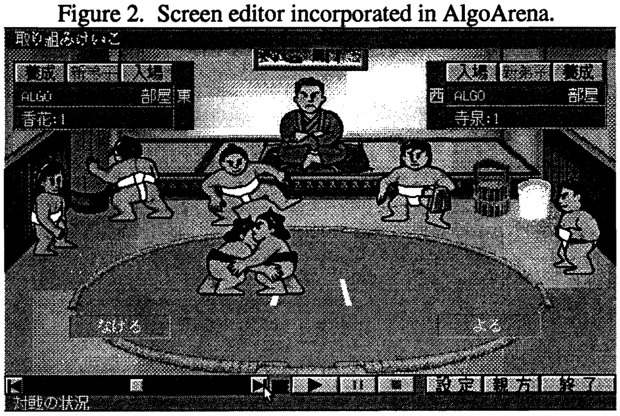

Figure 3. Fighting scene of AlgoArena.

```
REPEAT 30[
   IFELSE (:_my_position = 8)
      [move_forward]
      [IFELSE (:_his_posture = 4)
         [slap_down]
         [push_forward]]]
```

Figure 4. Example of a program.
(Some parts translated into English)

AlgoArena is a simulation of a sumo bout. A student (player) has his/her own wrestler, and makes a program describing the wrestler's actions with a screen editor, as shown in Figure 2. The editor is designed so that beginners can enter almost all the reserved words (commands, special symbols, system variables, and so on, as shown in Table 1) just by clicking on the menu. The student then lets his/her wrestler fight with opponents programmed by other students or teachers. The fighting animation is seen on the graphic screen of the monitor (Figure 3). The student is supposed to analyze the fight, find out the causes for defeat, devise tactics, and incorporate these into the program. Another cycle of activities is regenerated by a return match. Through these cyclic activities, students are expected to develop comprehensive capabilities for the basics of software design.

Figure 4 shows an example of a program. The program repeats the following thirty times: if my wrestler is on the edge of the ring (`my_position = 8`), let him go forward; if not and the opponent is bent far forward (`his_posture = 4`), then slap the opponent down; otherwise push the opponent forward.

The game of AlgoArena continues as follows. Either of the programs runs, referring to the current states of the game, determines the subsequent action command, and stops when this is determined. The other runs in the same way. When a pair of action commands is determined, it is evaluated at once. This is so that the first mover will not have a (dis)advantage. According to the combinations of the commands and the current states of the game, the game status change. The match ends when the conditions of the two wrestlers are such that one of them wins. If a match is not won by either wrestler after thirty sets of action, it is declared a draw.

The system of the game is definite, i.e., no element of chance is allowed to affect the outcome of the match. Therefore, the same match always follows the same process, and a match between identical wrestlers never fails to end in a draw. Since there is little chance to win by accident, deep thought is required for a sure win. Consequently, as student's capabilities in software design improve, the wrestler gets stronger.

2.3. Experience from an exploratory case study

The authors ran a pilot experiment at a municipal junior high school. The subjects were 27 students, aged from 13 to 15, and classes were held once a week for 6 weeks; each class was 50 minutes long. Most of the class time was spent making programs. Oral explanations were principally limited to system operation and sample program behavior. Apart from this, a manual of programming commands was distributed. Students were expected to learn by talking with others, as well as by looking at sample programs.

According to the answers to the questions after the classes: "What was interesting in AlgoArena?", the most frequent answer (41%) was "to have matches with friends." The answer to the question: "Who do you prefer to fight with?" was in 19% of cases 'programs made by peers', 4% preferred 'pre-installed sample programs', and 52% preferred 'both of them'. These results show that students were mainly motivated by social interaction with others. In addition, in the answer to the question: "With whom did you consult on programming? Name them.", 78% of students named consultation with peers. This result and our observation show they actively communicate with one another. Consequently, it seems plausible that a community of students, in which they are keen on problem-solving activities toward a common goal, was successfully formed.

Regarding performance of learning, almost all (96%) understood the usage of `REPEAT` and `IF` correctly. In the junior high school, BASIC language class is mandatory in the 9th grade and one of its goals is for students to understand the usage of `FOR..NEXT` and `IF` after 9 to 10 class hours. Compared with this, almost all students using AlgoArena reached this level in a shorter (less than 2/3) time. In particular, the class teacher reported that he was surprised to see a certain student of the lowest ability working on the activities very aggressively, and he created a program of nearly average level.

3. Conclusion

The design principles and system overview of AlgoArena are presented in this paper. The game situation incorporated into AlgoArena can provide students with a social setting that they commit to problem-solving itself at their own risk. It is comparable to the authentic activities of programmers. In this sense, activities in AlgoArena have high social reality. The exploratory case study revealed that a community of practice was able to be formed successfully and students enjoyed the social interaction with others. Moreover, learning performance reached the same level as that for conventional methods in a shorter time.

The authors will look into more cases to examine interaction among students and how they develop their community of practice.

References

1. Brown, J. S., Collins, A. and Duguid P. 1988. "Situated Cognition and the Culture of Learning", IRL Report No. IRL88-0008.

2. Kato, H. and Ide, A. 1993. "AlgoArena: a System for Learning Programming through Social Activities", Proceedings of 4th European LOGO Conference, pp.239-246.

3. Lave, J. and Wenger, E. 1991. Situated Learning — Legitimate Peripheral Participation —, Cambridge University Press.

Authors' Addresses

Hiroshi Kato and Akiko Ide: Applied Information Technology Research Laboratory, Information Technology Research Laboratories, NEC Corporation, 1-1, Miyazaki 4-chome, Miyamae-ku, Kawasaki, Kanagawa 216 JAPAN. {kato, akiko}@joke.cl.nec.co.jp.