An Authoring System for Creating a Practice Environment in the Network Service Field

Minoru Kiyama and Yoshimi Fukuhara
NTT Network Information Systems Laboratories
1-2356, Take, Yokosuka, Kanagawa, JAPAN
E-mail: kiyama@nttkb.ntt.jp
FAX: +81-468-59-3633

Abstract.

This paper describes an authoring system whose main purpose is to reduce the cost of developing and maintaining courseware which contains procedural knowledge used in the network service field. This aim can be achieved by considering the characteristics of this field. Material knowledge is divided into two parts, behavioral knowledge and procedural knowledge. We show that both of these parts are constructed by an easy authoring methods and efficient modification algorithms. This authoring system has been used to build several types of courseware, and the development costs have been reduced.

1. Introduction

The work presented here concerns a practical authoring system for training in the network service field, which is different from the authoring system used in schools. In our company, which is a supplier of network communication services, there are more than 200,000 employees engaged in maintenance, troubleshooting and sales of network services. Therefore expert training is very important. Network services personnel need many types of useful and interesting courseware for studying a lot of equipment and a variety of businesses, but the authors have limited time to build courseware. Why aren't intelligent tutors, by which students can give effective and useful instructions, being used in industry? One of the main reasons is the lack of practical intelligent development tools[1]. We developed an authoring system for the scene-based intelligent Computer Assisted Instruction (CAI) system in order to learn declarative knowledge, such as, what are the concepts and architectures of the switching systems[2,3,4]. The usefulness of the courseware built by this system was confirmed. It is also important for people who work with network equipment to be trained in procedural knowledge, such as how to operate digital switching systems, as well as declarative knowledge. It is known that a practice environment, based on direct manipulation, including interactive digital video for the sake of reality, helps people to effectively acquire procedural skills. We are particularly interested in a practical authoring system for tutoring procedural knowledge of many kinds of domain.

Another aspect of our work concerns the differences between the authoring system for the industrial field and the one for schools. Authoring systems must reflect the characteristics of the field that is applied. Physics laws, in general, are stable domains. The contents of such courseware are not changed, once the courseware is well established. On the other hand, many kinds of network equipment described in courseware are continually undergoing slight changes, and therefore the courseware itself needs to be frequently changed. In most cases, it is not necessary to set a deadline when building courseware concerning a stable domain, such as physics. The usefulness of that material will not decrease with a passing of time. In the network field, setting a time limit is important, because a courseware's usefulness might reach its peak before and after installing new equipment. The courseware's value decreases sharply after that time. It is useful for authors to have an authoring system that can modify courseware very simply and easily, and can develop courseware rapidly.

This paper is organized as follows. Section 2 presents the requirements for authoring systems in the network service field. Sections 3 and 4 explain how to build behavioral knowledge and procedural knowledge, respectively. Then in section 5, the development costs of our system are evaluated. Finally, we conclude with current status of the system.

2. Overview of Our System

2.1 Characteristics of network service field

It is important to make clear the characteristics of the network service field from the viewpoint of training when the authoring system is designed. These characteristics should be considered from the supplier side, not the users side.

a) Large scale - many kinds of complex equipment -
Supplying network services requires providing hundreds of kinds of complex communications equipment, such as digital switching systems, automobile telephone systems, ISDN systems and so on. The kinds of equipment are increasing as we begin to provide new services and old equipment still remains for a long time, until it is depreciated. In the case of switching systems, crossbar switching systems have been installed for more than a decade and more than ten kinds of digital switching systems are now being used.

b) Procedural knowledge
It is very important for employees to learn how to operate the equipment, as well as know its concept and how it is constructed, for the sake of troubleshooting and daily operations. Most equipment is operated by turning on/off switches on the panels, pointing to the abstract figure drawn on the display, and typing a command like a character-based command of UNIX.

c) Non-stop
Network systems are the infrastructure of a nation. Therefore, human error must not be the reason for a system being down. Novice operators can not use the actual machines in order to learn the operations.

d) Area configuration
Network services cover the nation, but equipment is installed distributively in several local areas. Different environments, such as the number of customers and the amount of communications traffic, cause different system configurations. Although the basic operations of the devices are the same in each region, some detail procedures are slightly different.

e) Frequent changes in equipment and operation
Many kinds of network systems are in general use for a long time, but they are continually undergoing slight changes in order to improve the systems. Part of the equipment or some of its software may change, therefore the appearance of the equipment and the order of the operation sequences may also change. The changes in the environment may also affect the operations.

f) Timely training
The training of equipment operations should be undertaken before and after introducing new equipment. The life cycles of equipment and training are depicted in Figure 1. The value of the training drops sharply after the peak.

g) Who is author
This characteristic is related to the organization of the company, and not with the equipment itself. Most courseware is built by operators and instructors rather than designers of the equipment. These authors do not have detail knowledge of the equipment and do not usually have any programming skills.

2.2 Requirements for the authoring system

We identify some requirements for the authoring system, which reflect the above characteristics in the network service field. Some characteristics may cause other requirements. These will be handled by setting the priority and trade-off of the conditions. Figure 2 shows the relationship between these characteristics and the requirements.

1) Practice environment
It is very important for people who work with network equipment to learn procedural knowledge, as well as declarative knowledge. The authoring system needs to build a practice environment for people

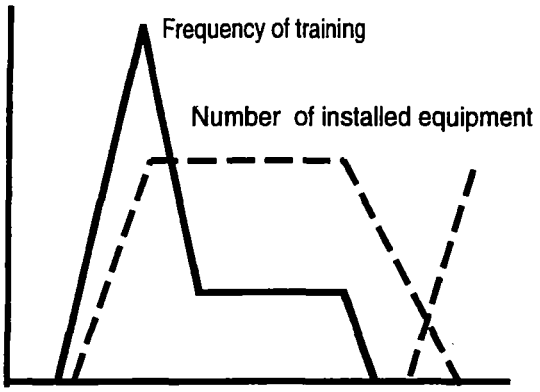


Figure 1. Lifecycle of equipment and training

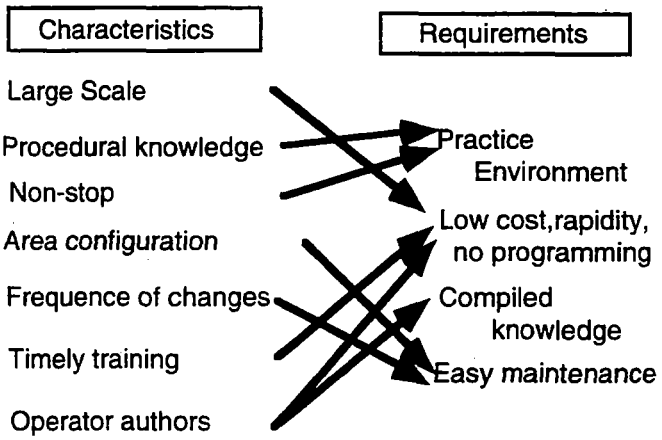


Figure 2. Relationship between field characteristics and requirements for authoring

to be trained in procedural knowledge. We have developed another authoring system for the scene-oriented intelligent CAI tutor for learning declarative knowledge, because these two types of knowledge require different types of instruction.

2) Low cost, rapidity and no programming
Courseware is built by employees who work with equipment rather than design the equipment. Thus, a practice environment should be created without programming. Many kinds of equipment requires spending extra time building different courseware and providing the courseware before the peak requires a rapid authoring system.

3) Compiled knowledge
Courseware authors do not have detail knowledge of equipment, like designers. Our system's aim is that logical domain knowledge, except for real-domain-knowledge, can be constructed by using the knowledge described in a users manual or an operations guidebook. Authors can supplement the lack of logical knowledge by using real-domain-knowledge, such as graphics, voice, motion pictures and so on, because they can see the behavior of the equipment very well. So the quality of the courseware will not be so bad.

4) Courseware maintenance
Many kinds of equipment described in the courseware are continually undergoing slight changes, and therefore the courseware itself needs to be frequently changed. The authoring system must have the capability of modifying courseware very simply and easily. The area configuration and the growth of that configuration determines the requirements for courseware maintenance.

2.3 Outline of system architecture

First, we show the operation model of the equipment in the real world, depicted in Figure 3. When the equipment receives a stimulus, such as trouble or manipulation, from other subsystems that are related to the equipment, the operation goal is fixed and the operator manipulates the equipment for the goal. An expert operator might assist them depending on the circumstance. Figure 4 shows the architecture of our system, which reflects the operation model in the real world. The material knowledge consists of three parts; simulation knowledge, training knowledge and real-domain-knowledge. Simulation knowledge represents the behavior of the equipment and enables the system to act in the same way as equipment in the real world. The tutor having that knowledge, will respond by either correcting or not correcting operations blindly. Training knowledge represents the procedural knowledge of the operations and manages the operation goal and stores the correct sequence of the operation to be learned. Real-domain-knowledge is the information to be presented to students and includes text, graphics, voice and motion pictures. Simulation knowledge, training knowledge and real-domain-knowledge reflect the equipment itself, the external stimulus and the appearance of the equipment, respectively, in the operation model

in the real world. Our authoring system consists of three parts for each domain knowledge. The tutoring system, which executes the simulation and training knowledge, is the interpreter for unary representation. Some knowledge is transformed with the instruction mode which is specified by the students. These architectures satisfy the requirements for the authoring system. The reasons and the details are shown in the following sections.

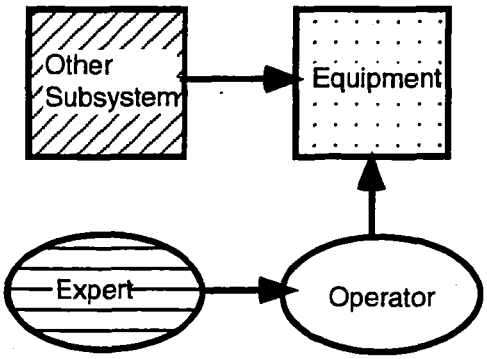


Figure 3. Operation model in the real world

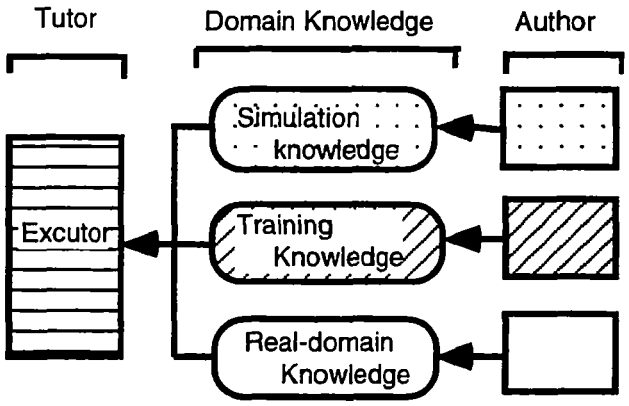


Figure 4. System architecuture

3. Authoring for Simulation Knowledge

3.1 Design concept

Simulation knowledge enables the system to behave in the same way as equipment in the real world, i.e. to blindly respond to either correct or incorrect operations. We have adopted the state transition model (STM) to represent simulation knowledge. This model is useful in describing the behavior of the network equipment. The steps of most operations are discussed below. Operators manipulate the equipment, then the equipment provides a response which the operators can see, and the internal state of the equipment changes to another state. These operations are represented exactly in STM. STM is also useful in describing the behavioral knowledge in the users manual, in which the sentences are written in a set of short pieces of knowledge, such as itemized form. We can represent the compiled knowledge, because STM covers both detail knowledge and rough knowledge in its representation ability.

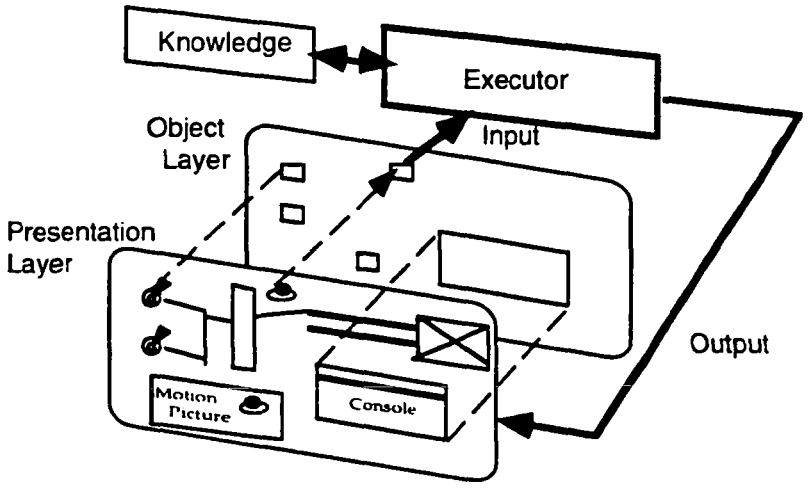
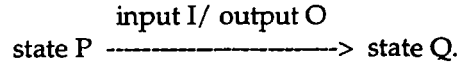


Figure 5. Layer of simulation knowledge

In the network service field, graphical user interfaces on the display are constructed by non-standard widgets, such as, network routes maps, as well as standard widgets such as a menu and dialog boxes. Two approaches exist for authoring the non-standard widgets: providing the object-base graphic editor and dividing the widgets into the transparent object layer and the presentation layer, like HyperCard (Figure 5). The object-based graphics editor in STEAMER has large sets of icons specifically designed to depict objects and represent behavior of objects in the propulsion domain[5]. It is clear that this editor will reduce the cost of authoring and provide easy modification to the courseware. However, we must implement an object-based editor for each domain. The cost is expensive and it takes a relatively long time to develop. We feel this approach is not suitable for rapid authoring and low-cost courseware development of many kinds of equipment.

3.2 Basic representation

We describe the simulation knowledge in the form of STM. The element, tuple is presented (P, I, O, Q), and is also visualized as follows;



This element means that the equipment receives input I at state P, then responds to the operator showing output O, and finally changes to the next state Q. We explain the implemented function in our system as follows.

a) Input: consists of object and event. The object is the target for the input. We have implemented two types of objects; a button object and a window object, which represents the button on the panel or the display and the character-based monitor console, respectively. The event is the stimulus given by the operators. The mouse and keyboard are used for input events.
b) Output: means the response to the operator from the equipment. The main functions are presenting graphics, voice, motion pictures on the presentation layer and setting the activity of group states.
c) State: represents the logical state of the equipment.

Additionally, you can place the objects on a motion picture on the presentation layer. The life of the object should be specified in this case.

3.3 Extended representation

When we try to describe the behavior of complex equipment in the real world, a lot f states may have to be defined. The equipment consists of several subsystems, and compound states related to some subsystems, which cause a number of states. Compound states, p1-q1, p1-q2, p2-q1,p2-q2, are the combination state of each subsystem when state p1 and p2 belong to subsystem A and state q1 and q2 belong to subsystem B. We have extended STM by adding the concept of group. A group is formed from a set of elements that have closed state transition, and therefore the compound state can be reduced. As a result, we can represent the behavior economically. Any inputted events are distributed to each group equally and transition occurs in each group independently. Although each group acts independently in state transition, the objects can be shared among the group. An event for shared objects may cause several transitions in each group. The order of firing, such as firing all of them simultaneously or no firing or firing the latest active state and so on, can be defined as simulation knowledge.

3.4 Tools

We provide some tools for entering the elements of STM and debugging the behavior. Two kinds of user interfaces are implemented for inputting: a table-formed interface for inputting the tuple form and a graphical interface for inputting the connection of states. You can chose either of them depending on the situation. The debugging tools execute the static check of contradiction and duplication, such as removing the isolated loop without an end state, and dynamic check of the behavior, such as, the traces of transition and execution on the way to the transition sequence. Our tutor executes unary representation, extending STM.

4. Authoring for Training Knowledge

4.1 Design Concept

Because tutoring systems with only simulator knowledge will respond to either correct or incorrect operations blindly, students can not take pedagogical instruction. When operators manipulate the equipment in a situation, they have some goals such as solving some troubles and have the plans and procedures in order to reach the goals, such as what sequences to use. It is important for students to learn the plans and procedures. Our authoring system stores the procedures as training knowledge, which is separated from simulation knowledge, i.e both kinds of knowledge are managed independently. The outline of authoring for the training knowledge is discussed below. Training knowledge is represented as the input sequences of STM which describes the simulation knowledge. Authors can build training knowledge by executing the process shown in Figure 6. First of all, authors define simulation knowledge and construct real-domain-knowledge when they build the courseware. After this, they can manipulate the display in the same way as with real equipment. Next, they enter the correct sequences of a procedure on the display by means of executing STM represented as the simulation model. The authoring system records the log of input as training knowledge. Our executor interprets only unary representation STM, so training knowledge is transformed into STM before the student starts to learn.

The merits of dividing the material knowledge into two parts are described below. i) Building training knowledge by using simulation knowledge will reduce the cost of developing the courseware, ii) We can add the procedures to be trained step by step without affecting the simulation knowledge. This facilitates easy maintenance of the courseware. iii)When some parts of the courseware are modified, both types of knowledge can be modified independently in some cases.

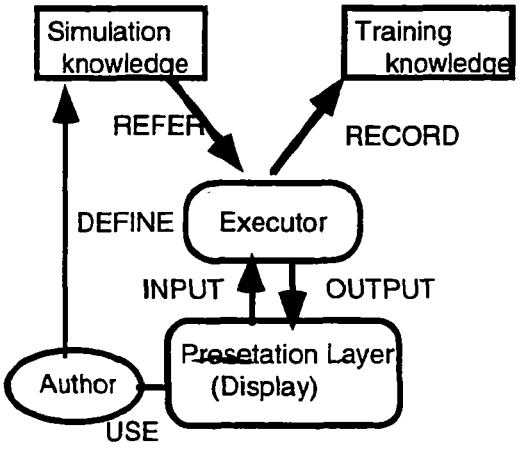


Figure 6. Building the training knowledge

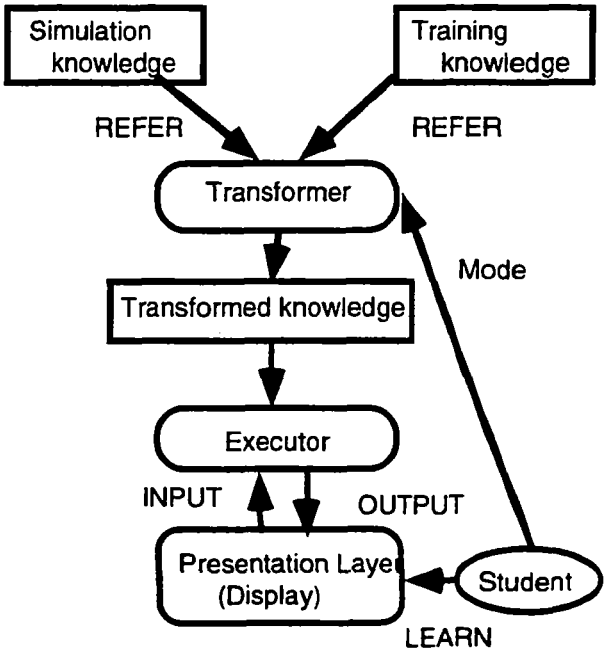


Figure 7. Transforming the training knowledge to STM

4.2 Transformation algorithm

The transformation algorithms from the training knowledge into STM are shown below. When a student starts to learn, they choose an instruction mode: in the imitation mode the system shows the student the correct operations, in the guide mode the student follows the operation guided by the tutor, and in the practice mode the student

manipulates freely. The transformations are accomplished for the mode which the student has chosen. We show a simple algorithm in the case of the imitation mode, although similar algorithms can be adapted in other cases.

The following is given; simulation knowledge T: {t1, t2, t3,...,tn}, tj = (pj, ij, oj, qj), training knowledge L: {l1, l2, l3, ... lm}, lk = (ik, pk), Confirmation input ic. For all elements lk of set L, find the tuple tj in T which satisfies the following condition, then rewrite the tuple of T. The set T' including rewritten tuples constructs the transformed training knowledge.

[condition] ik = ij of tj and pk = pj of tj.

[rewriting rule] tj is rewritten into (pj, ic, o'j, qj) , in which o'j is oj added to the two outputs, blinking the next object ok+1 and inactivating the group in case pk and pk+1 belong to a different group.

4.3 Maintenance method

When the position of a button is changed on the real equipment, the authors redraw the button on the presentation layer and simply move a button object on the object layer. It is not necessary to modify any STM elements and training knowledge. When changing an equipment's appearance, such as color or form of parts, no domain knowledge needs to be modified, only the presentation layer. When the order of the operations is undergone changes, such as reversing the sequences, only the training knowledge need to be rebuild. No modification to the simulation knowledge needs to be made. Additionally, training knowledge can be added without affecting the simulation knowledge. On the other hand, the authors must modify all types of knowledge when a button is removed. They must delete the button on the presentation and object layers, and modify the simulation and training knowledge. The system shows the authors some parts of the simulation and training knowledge that must be modified, but it is author's responsibility to correctly modify the knowledge.

5. Development Cost

The cost of developing the courseware using the authoring system has been compared with the cost of programming. Figure 8 shows the development cost of the automobile telephone system courseware. The man-hours for creating logical knowledge is 15 percent that of the programming cost, although the cost of creating real-domain-knowledge is the same in both cases. The total development cost using our system, which includes building real-domain-knowledge, is a quarter of the cost of building the courseware using programming.

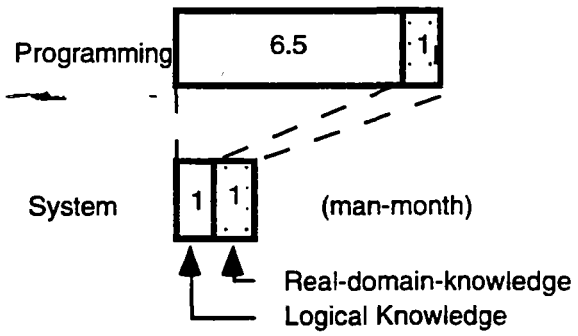


Figure 8. Authoring cost comparison

6. Conclusion

We have developed a practical authoring system for the network service field. The authoring system has several facilities which provide low-cost, rapid authoring. This system makes it easy to build and change procedural knowledge courseware. Simple and efficient methods are shown, that reduce the development and maintenance cost. The system was used to build large courseware, and the development cost was reduced to a quarter. This

system is working on personal computers. Many operator authors are building more than a hundred types of courseware in a year. A tutoring system was introduced into our three hundred branches and five training centers. It is important to balance the tutoring ability and the authoring cost when a practical authoring system is developed.

Reference

1. Woolf,B.P., Soloway,E., Clancey,W.J., Van Lehn,K. and Suthers,D., "Knowledge-based Environments for Teaching and Learning," AI Magazine, Vol.11, No. 5, 1991.
2. Fukuhara,Y., Kiyama,M. and Nakata,K., "Multimedia Authoring System for Practical Scene-oriented ITS (CAIRNEY)," Int. Conf. on Multimedia in Education and Training, 1991.
3. Morihara,I., Ishida,T., and Furuya,H., "Rule-based Flexible Control of Tutoring Process in Scene-oriented CAI Systems," Int. Conf. on Artificial Intelligence Applications, 1987.
4. Ishida,T., Sasaki,Y. and Fukuhara,Y., "Use of Procedural Programming Languages for Controlling Production Systems," Int. Conf. on Artificial Intelligence Applications, 1991.
5. Hollan,J.D., Hutchins,E.L., Weitzman,L.M., "STEAMER: An Interactive, Inspectable, Simulation-Based Training System," in Artificial Intelligence and Instruction, Kearsley G., Adison & Wesley, 1987.