협력적인 멀티미디어 시스템 디자인을 위한 도구 개발에 관한 연구

백 승 익†

요 약

기존의 연구에 의하면 멀티미디어 시스템(Multimedia Systems)을 성공적으로 디자인하기 위해서는 다양한 관련분야의 전문가들 사이에 긴밀한 협력이 필요하다고 한다. 그러나, 그 전문가들은 서로 상이한 관점과 지식 체계을 가지고 있기 때문에 상호간 긴밀한 협력에 필수적인 지식을 공유하고 의견을 조율하는데 많은 어려움을 겪는다. 특히 최근 들어 멀티미디어 시스템 디자이너들이 가상의 팀(Virtual Team)을 형성하여 인터넷과 같은 컴퓨터 통신시스템을 활용함으로 해서 지식공유와 의견조율을 통한 협력은 한층 더 어렵게 되었다. 본 연구에서는 이러한 문제점을 해결하는 Designer's Café을 개발하는데 그 주요 목적을 두고 있다.

Designer's Café : Developing A Tool for Collaborative Multimedia Systems Design

Seung Ik Baek†

ABSTRACT

Multimedia systems design generally requires a collaborative effort from a group of designers with a variety of backgrounds and tasks, such as content experts, instructional designers, media specialists, users, and so forth. The fact that the designers have totally different backgrounds and tasks may hinder their collaboration. This research develops a collaborative design tool for multimedia systems design, called the Designer's Café, with which multimedia designers can share their design knowledge on the Web freely. The Designer's Café consists of three intelligent agents that are designed for supporting collaboration activities of multimedia designers. The intelligent agents are implemented by using JAVA SCRIPT and COLD FUSION. By using the tool, designers create, revise, and share their knowledge/ideas in very structured ways. The structured design process might enhance the collaboration activities of designers while designing a multimedia system within a virtual working space.

키워드 : 멀티미디어 시스템 디자인(Multimedia Systems Design), 인텔리전트 에이전트(Intelligent Agents), 소프트웨어 디자인(Software Design), 디자인 도구(Collaboration)

1. INTRODUCTION

Multimedia is an umbrella term generally referring to the seamless integration of text, sound (such as spoken words, music, or sound effects), visual images (such as still photographs, motion pictures, or animation), or other electronically represented information under computer software control [1, 27]. Therefore, a multimedia design project generally requires a collaborative effort among many developers with a variety of backgrounds, such as personnel, context experts, instructional designers, users or clients, and so forth [2, 6, 25]. Because each of them possesses totally different backgrounds, training, and experiences, communication among them is always problematic. Unifying a design team that is comprised of members from different disciplines, with different skills and different ways of describing multimedia, might be a major problem that project managers must solve in order to complete projects successfully. Also, when team members are located at different functional groups within a firm or in separate firms, their collaboration becomes more difficult. The major issue in the collaborative design environment is making designers, with their own unique interests and individual perspectives, come to a common understanding of what they are building,

† 정 회 원 : 한양대학교 경영학과 교수
논문접수 : 2001년 8월 20일, 심사완료 : 2001년 12월 12일

sharing information, and coordinating their individual design efforts. While a great deal of research and theory exists for designing more effective multimedia applications with respect to content, structure, and presentation [12, 19, 20], very little attention has been paid to the actual coordination process that individual designers incur or groups undergo in a collaborative design process for multimedia systems [25]. And, design tools for multimedia systems currently on the market are mainly designed for a single designer, not multiple designers. The research aims to develop a tool, called Designer's Café (a tool for collaborative multimedia systems design) that can facilitate expression, transmission, and evaluation of ideas from design participants.

2. Collaborative Design

The design task involves a complex set of cognitive activities [8, 18, 22]. Because designers hardly have a complete statement of a design problem and a straightforward design process to follow, they use a variety of cognitive activities to analyze a problem and to develop a potential solution of the problem [18, 22]. Simon [23] states that "design is an ill-structured problem in which the start state is unspecified, the goal state is incompletely specified, and the transformation function is unspecified." Goel and Pirolli [7] define that "design consists of two components : logical and creative components." Goel and Pirolli [8] also state that "design is fundamentally mental, representational, and a signature of human intelligence." Cross [3] argues that "design ability is summarized as comprising resolving ill-defined problems, adopting solution-focused cognitive strategies, employing abductive or appositional thinking and using non-verbal modeling media." Although design is too complex an activity to be understood clearly, based on the studies that examine the design process in relation to a wide variety of artifacts, there has been a gradual understanding of the management of design process [13, 18] the structure of design problems [7, 8, 26] the nature of design activity [3] and the philosophy of design method [22, 23]. One of the popular domains for studying design is the area of software design [4, 9, 13, 28].

Software design is a problem-solving process of translating clients' and end users' underspecified and ambiguous knowledge into a structured description of a computer program that will satisfy their needs [4, 13, 29]. Software design tasks are too complex to be accomplished directly from the design problem statement that clients or users specify. Thus, during a design process a designer decomposes the ill-defined and ill-structured problem into a set of well-defined and well-structured subunits that are manageable and minimally interacting, and combines these design solutions [13, 22]. The decomposition and integration processes are central to the success of software design [13]. Since the kinds of knowledge that may be required for accomplishing both problem decomposition and solution integration processes are practically limitless, individual designers face cognitive difficulties (breakdowns) during design processes. The cognitive breakdowns are likely to produce incorrect design solutions. To reduce the cognitive breakdowns, users, designers, and developers must share their knowledge and skills central to the design of useful software [24]. Curtis et al [4] and Guindon [9] state that, due to the lack of knowledge and cognitive limitations, the cognitive difficulties (breakdowns) occur.

2.1 Lack of Knowledge

The knowledge required for software design can be divided into two categories : a) application domain knowledge [1] and b) knowledge of the design process [13, 22]. When designers lack application domain knowledge, they have difficulty understanding the problem itself. Rosson and Alpert [21] argue that some of the most expensive mistakes in software design projects can be attributed to the designers' failure to understand the problem situation adequately.

Design schema is a higher-order knowledge about a design process [13]. In a design process, previous experience is a major factor that can provide a design schema for the problem solution. For instance, if a designer has experience in solving similar problems, the designer can use the same problem solving process to solve new problems. To construct design schema effectively, designers need to store and categorize previous experiences in a proper way for future use [13]. By using software tools that can help designers to organize, access, and modify domain knowledge and design schema in the course of solving problems, designers can overcome this problem [9].

2.2 Cognitive Limitations

During a design process, due to capacity limitations of

short-term memory and the unreliable retrieval of relevant knowledge from long-term memory, designers experience cognitive limitations [9]. Because of the lack of adequate knowledge organization, designers frequently fail to retrieve correctly and to apply relevant knowledge throughout a design process [1, 13]. Even in small-sized software design projects, due to the limited cognitive capacities, designers have difficulties in considering all of the stated or inferred constraints required in generating a design solution. By using software tools and methodologies that allow designers to consider all the stated or inferred constraints in defining a solution, to keep tract of the status of subproblems, and to merge solutions from individual subproblems in order to form a complete solution, designers can overcome their cognitive limitations [9].

Although many tools have been developed to alleviate the cognitive breakdowns in large-scale, complex software design projects, these projects still have problems. Since the large-scale, complex software design typically involves knowledge across multiple domains and includes numerous, potentially interrelated constraints and assumptions, a single designer cannot possess all the knowledge and consider all the constraints and assumptions even if the designer uses tools [29]. In such an environment, the breakdowns can be considered to be the norm [28]. Therefore many large, complex software design projects are performed in a team environment [4, 15, 28].

Since the knowledge required for large, complex software design is unavailable from any one individual designer, software design team members integrate their mental models of potential design solutions into an effective system through formal and informal communications [15, 29]. These interactions help designers to overcome each other's cognitive breakdowns. However, eliminating cognitive breakdowns through these interactions requires effective informal and formal interpersonal communication channels for sharing knowledge among team members [15]. The difficulties of communication restrict individual designers in supporting others' cognitive breakdowns during a design process.

The use of multimedia has been popular in various application areas. Especially, in the area of education and training software, multimedia has an important role. Because multimedia systems design requires knowledge about media and instruction above the knowledge required for the conventional text-based software design, designers experience more serious cognitive breakdown [10]. Successful multimedia systems design requires an interactive design environment in which design participants, including users, can share their design solutions freely.

3. Multimedia Systems Design

Multimedia systems design consists of three basic cognitive activities, which are usually performed by a group of designers : selecting information, structuring contents, and selecting media [6]. By analyzing application domains and target audiences, designers decide what information should be contained in a multimedia system. Then, they identify a hierarchical relationship (content structure) among the selected information sources. Finally, they decide how the selected information should be presented in a multimedia system. These activities are very closely related, rather than separated.

Teamwork, cooperation, coordination, and communication among team members are very important in a collaborative design process for multimedia systems. Because multimedia systems design involves various kinds of knowledge at the different levels, the teamwork, cooperation, coordination, and communication are more important in designing multimedia systems, compared with other text-based systems. One obvious recommendation is to promote actively the acquisition, sharing, and integration of knowledge within a design team [28]. In order to enhance knowledge exchange and resolution of conflicts among team members, many real-world design teams construct a shared memory [28, 29]. Two aspects of group dynamics - knowledge exchange and resolution of conflicts - are vital to collaborative multimedia systems design. However, because many design teams manage the shared memory in an ad hoc manner, and just store formal knowledge, such as users' manuals and user analysis reports, they cannot manage the dynamic evolution of knowledge that occurs throughout a design process [28]. In order to manage and speed the evolution of knowledge, it is necessary to develop new tools that encourage design participants to bring their knowledge and skills to bear on the design. Also this should be accomplished without forcing them to mediate through ex-

plicit description. The research aims to design a tool that can facilitate expression, transmission, and evaluation of ideas from design participants.

4. Tools for Supporting Multimedia Systems Design

When designers first start thinking about a visual interface, they often sketch rough pictures of the screen layouts on paper rather than using a computerized design tool. Because their initial goal is to sketch the overall layout and structure of the components quickly, rather than specify detail interface design, they prefer to sketch early design ideas on paper or a whiteboard. A variety of computerized design tools have been developed to support the paper sketch. Unlike the paper sketch, the computerized design tools allows designers to sketch their initial ideas interactively and modify their sketches easily, as well as to produce sketches quickly and flexibly. In many tools, the screens, which designers quickly sketch, are often tied together by storyboarding techniques [14]. The storyboard shows how one frame relates to adjacent ones, as well as to the whole concept, and what media are needed to achieve a particular effect. In the early design stage, designers need tools that give them the freedom to sketch rough design ideas quickly, the capability to specify transitions between screens and behavior of interface elements, and the flexibility to fill in the design detail [16].

SILK [16] is an electronic sketching tool that allows designers to quickly sketch an interface electronically. By using SILK, designers sketch a series of storyboards and illustrate transitions among them. Beyond providing the ability to capture design ideas quickly, SILK allows designers to examine, annotate, and edit their sketches easily throughout the design process.

DENIM [17] is an electronic sketching tool for early web interface design. DENIM is most closely related to SILK [17]. DEMIN takes many ideas from SILK and extends them to the web design. Both SILK and DEMIN are storyboard-based systems that allows designers to quickly sketch initial design ideas in the storyboards, create links among storyboards, and interact with them in a run mode. However, DENIM de-emphasizes the screen layout aspects of SILK, focusing on the structure of whole screens [17]. DENIM provides the different ways of viewing storyboards, from individual storyboards to a whole picture.

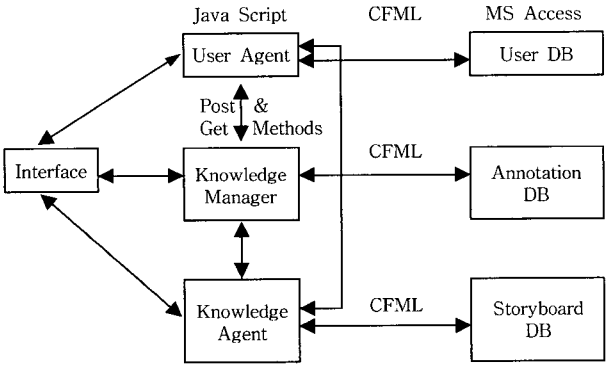
Anecdote [11] is a multimedia design tool providing supports for the early early-design phase as well as the whole development process of multimedia systems. Like DENIM and SILK, Anecdote enables designers to sketch design ideas by creating storyboards and connect them with navigational links. Similar to DENIM, Anecdote allows designers to explore storyboards at the different levels of abstraction. However, in supporting multimedia design, SILK and DENIM might have problems. Multimedia designers need to explore creative temporal layouts, more expressive interactions, and the use of dynamic media such as audio, video, and animation within an application. To satisfy the needs of multimedia designers, Anecdote employs the concept of surrogate media and surrogate scene. Surrogate media is a type of media to represent the content of the media data to be created. Anecdote helps multimedia designers to sketch not only the screen layouts, but also media contents included in screens. Surrogate scene is a design sketch of a screen representing a segment of the application structure. Anecdote employs a top-down design approach. This is, it allows designers to sketch the scenario roughly, and later helps designers to refine the details of the scenarios.

Designing a multimedia system can be characterized as a learning, communication, and negotiation process among design team members [2, 6]. Throughout the interactive process, the team members integrate knowledge and skills from designers in various domains. None of the design tools support team activities. The objective of the research is to develop a collaborative support tool for multimedia systems design, called Designer's Café. Like SILK, DEMIN, Anecdote, Designer's Café is a storyboard-based design tool providing supports for the early design phase. Similar to Anecdote, Designer's Café employs the concept of surrogate media and surrogate scene, and is specialized in multimedia systems design. However, it supports team design activities, as well as individual designers' activities. Designer's Café supports team dynamics in the process of multimedia design. Designer's Café enables design team members to explicitly communicate and negotiate how narration, images, text, special effects, and background music/images are brought together and linked together to form a final presentation. By helping designers create, exchange

and share storyboards, we can enhance their knowledge communication and sharing activities. This research designs a collaborative support tool that can manage diverse knowledge sources and their dependencies throughout the multimedia design process.

5. Designer's Café : A Collaborative Design Tool for Multimedia Design

An intelligent collaborative design tool for multimedia systems design, called Designer's Café, consists of three intelligent agents : user agent, knowledge manager, and knowledge agent. The intelligent agents are designed to support the collaboration activities of designers. Each agent consists of a knowledge base and a set of production rules that manage its knowledge base and generate adequate interfaces for designers. In order to perform their tasks, the agents communicate with other agents through HTTP communication protocols. An agent might request information from other agents. The knowledge bases are implemented using Microsoft Access. The rules are implemented using Java Script. For the interface between Java Script and Access, we use the CFML (Cold Fusion Markup Language). The overall system architecture is shown in (Figure 1). By communicating with each other, agents dynamically generate the interfaces. <Table 1> summarizes how intelligent agents can support a collaborative multimedia design.



(Figure 1) Overall Architecture of Designer's Café

(Figure 2) shows a sample code of an intelligent agent. When agents are loaded, they read their databases to update their knowledge (Part 1). Whenever changes are detected in any databases, all three agents are reloaded. Based on their knowledge, agents generate proper interfaces for designers (Part 2). If agents need more information from other agents, they formulate queries (Part 3).

〈Table 1〉 Roles of Intelligent Agents

Collaborative Activities	Roles of Intelligent Agents
Knowledge Creating	Intelligent Knowledge Creating Templates • Big Picture View • Remind/Cue • Automatic Agenda Management • A Structured Knowledge Creating Mechanism
Knowledge Storing	Intelligent Indexing and Structuring Mechanisms • Social Indexing • Issue-Based Indexing
Knowledge Distributing	Intelligent Knowledge Distributing • Various Views of Changes • Filtering Changes • Dynamic and Automatic Knowledge Distribution
Knowledge Retrieving	Intelligent Knowledge Retrieving Engine • Content-Based Retrieving • Context-Based Retrieving • Automatic Query Generation

```
<CFQUERY NAME = "GET_SB"  DATASOURCE = "SB">
        SELECT * FROM STORYBOARD
        WHERE ID = # URL_NO #
</CFQUERY>

<HTML> <BODY>                                          PART 1

<SCRIPT LANGUAGE = "JAVASCRIPT">
.
Function DisplaySB(no)
{
        <CFOUTPUT QUERY = "GET_SB">
        if (GET_SB.ID == no)
        {       doc.document.write('<b>…</b>') ; …     PART 2
        }
        </CFOUTPUT>
}

if (UserRequest == "ShowSB") {DisplaySB(id) ; }
.
</SCRIPT>

<FORM NAME = "REQUESTANNOTATION" ACTION = "KMGR.
 CFM?NO = # ID # & USER = # NAME #" METHOD = "POST">
.
</FORM>

</BODY></HTML>                                         PART 3
```

(Figure 2) Sample Code

5.1 User Agent

The main functions of the User agent include :

- Remember all design activities while designers use the Designer's Café. While designers use the Designer's

Café to share their design knowledge, the User Agent keeps tracking their collaboration activities. It helps designers monitor their own design activities.

- Dynamically organize designers' agendas. The User Agent creates, modifies, deletes, and prioritizes agendas. Whenever designers get into the Designer's Café-based design environment, it creates, modifies, and deletes agendas repeatedly based on the changes that are occurring in the databases of the Knowledge Agent and the Knowledge Manager. To detect the changes in the databases of the Knowledge Agent and the Knowledge Manager, the User Agent continuously communicates with them. In addition, the User Agent prioritizes agendas based on the pre-defined rules. For example, when a designer makes comments on a design idea, the comments are shown in the top priority list of its originator.
- Provide templates for knowledge creation. The User Agent contains templates for creating knowledge and rules for validating user input in the templates. It has the templates for creating storyboards, hot spots (objects for user interactions), and annotations.

5.2 Knowledge Manager

The Knowledge Manager performs the following functions :

- Monitor all changes that occur in a knowledge repository and forward them to the User Agent. The Knowledge Manager monitors all changes that occur in storyboard database and forwards them to the User Agent. Based on the information, the User Agent generates and prioritizes agendas of individual designers.
- Reformulate the designer's queries based on the designer's need. If designers retrieve storyboards for review, the Knowledge Manger generates a query to retrieve all related information. If designers want to retrieve the similar storyboards by a keyword, the Knowledge Manager automatically generates two different queries for searching the storyboard database and the annotation database. Once it retrieves all necessary information, the Knowledge Manager generates an interface to present the information effectively.

5.3 Knowledge Agent

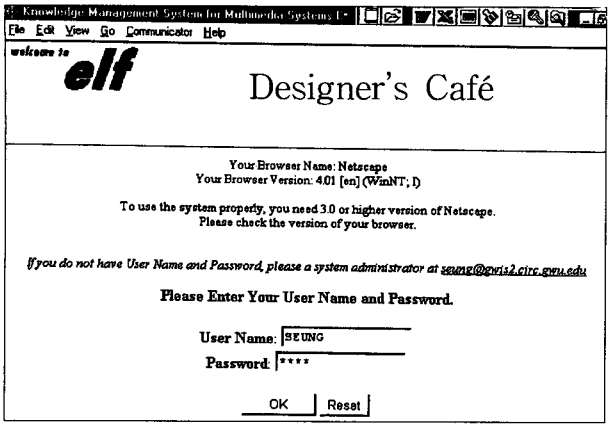
The primary goal of the Knowledge Agent is to maintain the integrity, accuracy, and quality of storyboards. A mechanism is needed to identify and promote the most useful knowledge and to filter useless, outdated, or incorrect knowledge from a knowledge repository. This is especially important in collaboration on an open network environment, like the Web. The Knowledge Agent performs important functions in implementing the mechanism.

- Index design knowledge. When new knowledge is created, the Knowledge Agent automatically indexes and saves the knowledge.
- Detect inconsistency and generate recommendation. When designers create new storyboards or revise existing storyboards, the Knowledge Agent checks their inconsistency and duplications with the existing storyboards. Since the Knowledge Agent has knowledge about the relationships among different attributes within the templates for the knowledge creation, it can check consistency and duplication of new knowledge and can recommend the resolution techniques when it detects inconsistency or duplication.

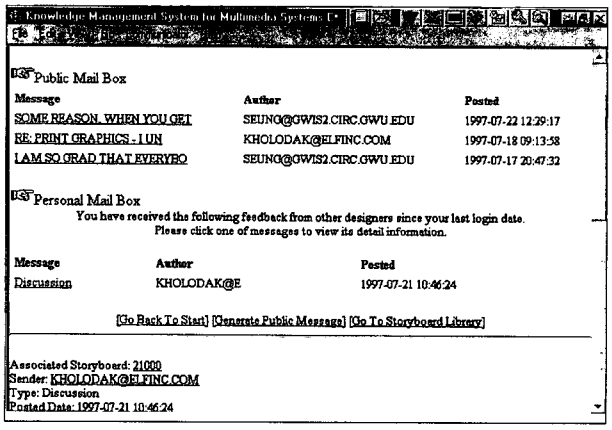
5.4 Implementation of Intelligent Agents

This section presents a step-by-step description of how a designer works in the Designer's Café-based collaboration environment. (Figure 3) is the initial screen of the Designer's Café. When a designer enters his/her user name and password in the initial screen, the User Agent records his/her login date and time. Then the User Agent sends a query to the Knowledge Manager and the Knowledge Agent to retrieve the storyboards and the annotations that have been modified since the last login date and time. Once it receives all information, it generates and prioritizes the agendas. (Figure 4) shows the mailbox of a designer. The mailbox consists of the public mailbox and the personal mailbox. The public mailbox contains public announcements. Additionally, the personal mailbox contains the agendas that designers need to perform. If other designers comment on a storyboard or change a storyboard, its originator gets an urgent message in his/her personal mailbox. The User Agent manages the mailboxes.

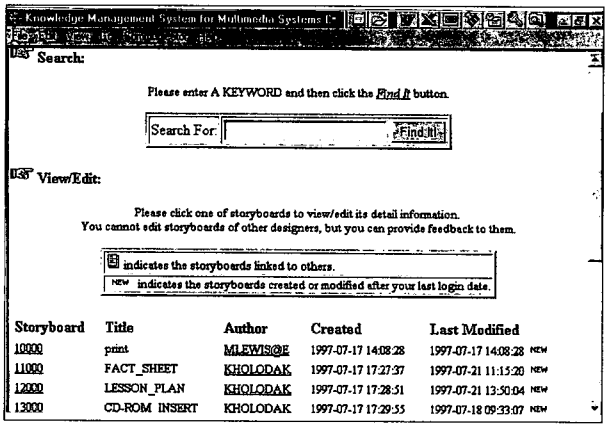
After checking the agendas, a designer goes into the main screen for viewing the storyboard library (Figure 5). The Knowledge Manager and the Knowledge Agent create the screen. From the screen, a designer can actually start



(Figure 3) Login Screen

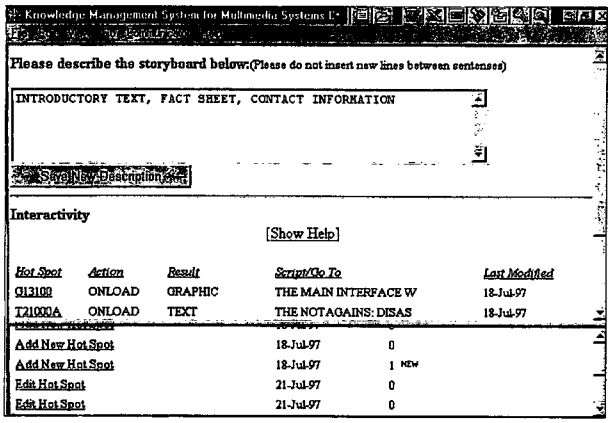


(Figure 4) Agenda Screen

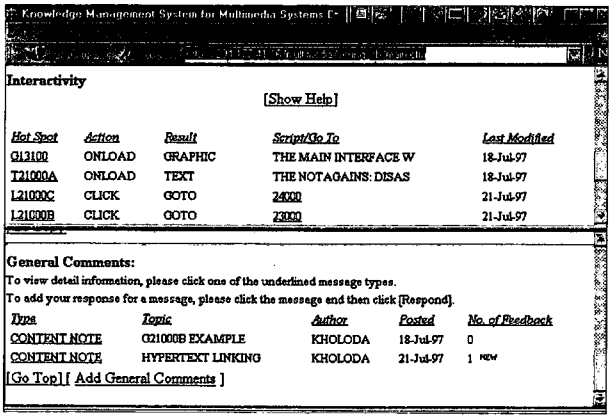


(Figure 5) Main Screen

his/her design activities. By entering a keyword, he/she can retrieve the storyboards and the annotations related to the keyword. The Knowledge Manager generates a query to retrieve all information that is related to the keyword. By clicking one of storyboard numbers, he/she can view the detailed information about the selected storyboard (Figure 6). The storyboard consists of a description and hot spots. The description provides general information about what information the storyboard presents, and how the information is presented in the storyboard. The hot spots are the objects that users can interact. In (Figure 6), the lower frame keeps tracking the storyboard changes. When a designer changes a specific storyboard, the Knowledge Manager accumulates the changes and shows a list of changes when the storyboard is retrieved. In a storyboard, a designer can also view its annotations and add new annotations (Figure 7).



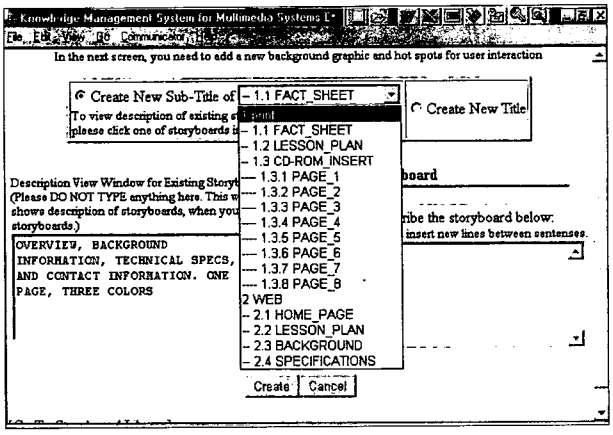
(Figure 6) Storyboard Screen & Change Tracking Screen



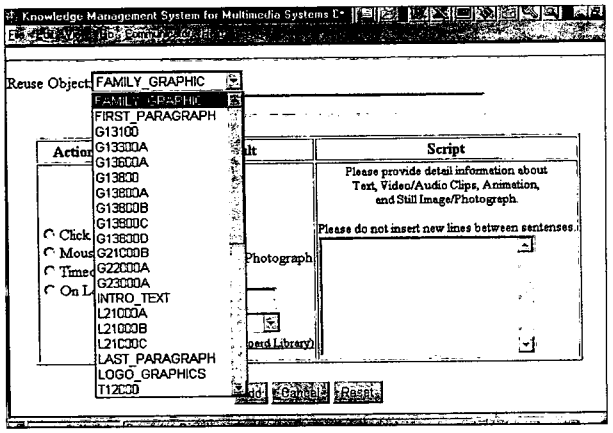
(Figure 7) Storyboard Screen & Annotation Screen

The User Agent has three different templates for creating storyboards, hot spots, and annotations. (Figure 8) shows a template for creating storyboards. A designer can create either a super-storyboard or a sub-storyboard under a super-storyboard. By filling out the title and the description of a storyboard in the template, he/she can create a new storyboard. (Figure 9) shows a template for creating hot spots. By clicking one of the hot spots in the library, a designer can reuse the hot spots. He/she also can create

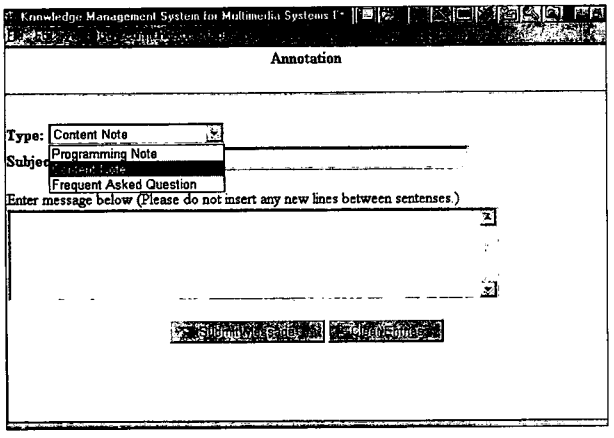
new hot spots. Based on the created contents, all serial numbers of storyboards and hot spots are generated by the Designer's Café (Figure 10) shows a template for creating annotations. For the purpose of indexing, three different types of annotations are identified : programming note, content note, frequent-asked-question (FAQ).



(Figure 8) A Template for Creating Storyboards

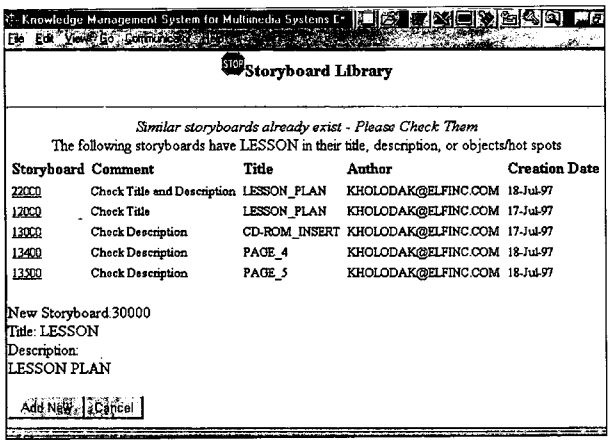


(Figure 9) A Template for Creating Hot Spots

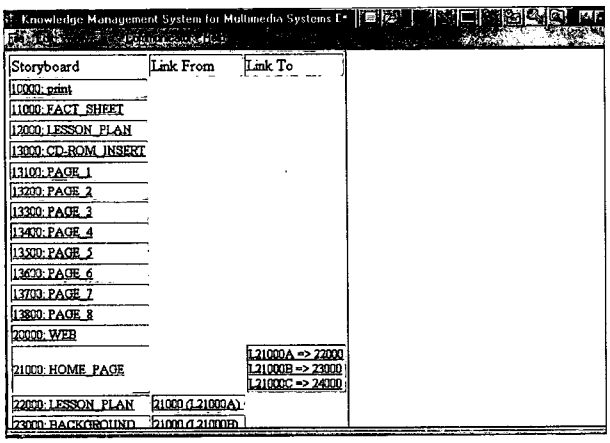


(Figure 10) A Template for Creating Annotations

Before a new storyboard is stored in the knowledge repository, the Knowledge Agent checks its consistency with the existing storyboards. If it detects any duplication and inconsistency, it provides recommendations to a designer (Figure 11). To show the relationships among storyboards, the Knowledge Agent generates a matrix (Figure 12). From this matrix, a designer can navigate various storyboards.



(Figure 11) Consistency Checking Screen



(Figure 12) A Matrix Screen for Viewing Relationships among Storyboards

6. Summary

By implementing three intelligent agents, we develop a collaborative design tool for multimedia systems design, called Designer's Café with which multimedia designers can share their design knowledge freely among themselves on the Web. By using the tool, designers create, revise, and share their knowledge in very structured ways. The structured design process might enhance the colla-

boration activities of designers while designing a multimedia system within a virtual working space. Without having the structured process, they might face difficulty in coordinating their design efforts. Designer's Café is a promising tool for collaborative multimedia design. We have implemented the Designer's Café to demonstrate the desired features of a collaborative design support tool. The next goals of this research are to develop the interfaces with multimedia authoring tools, such as HyperCard, Director, and to explore the usability of the Designer's Café in real-world settings.

References

[1] Adelson, B. and Soloway, E., "The Role of Domain Experience in Software Design," IEEE Transactions on Software Engineering, November, 11(11), pp.1351-1360, 1985.

[2] Alber, A. F., Multimedia : A Management Perspective, Wadsworth Publishing, Belmont, CA., 1996.

[3] Cross, N., "The Nature and Nurture of Design Ability," Design Studies, July, 11(3), pp.127-140, 1990.

[4] Curtis, B., Krasner, H. and Iscoe, N., "A Field Study of The Software Design Process for Large Systems," Communications of the ACM, 31(11), pp.1268-1287, 1988.

[5] Chapman, "Accelerating the design process : A tool for instructional design," White Paper of Designer's Edge, Allen Communication, 1995.

[6] England, E. and Finney, A., Managing Multimedia, Addison-Wesley, NY, NY., 1996.

[7] Goel, V. and Pirolli, P., "Design within Information-Processing Theory : The Design Problem Space," AI Magazine, Spring, pp.19-36, 1989.

[8] Goel, V. and Pirolli, P., "The Structure of Design Problem Spaces," Cognitive Science, 16, pp.395-429, 1992.

[9] Guindon, R., Krasner, H. and Curtis, B., "Breakdowns and Processes During the Early Activities of Software Design by Professionals," in G. Olson, E. Soloway and S. Sheppard (eds.), Empirical Studies of Programmers, Vol.2, Albex, Norwood, NJ, pp.65-82, 1987.

[10] Heller, R. S. and Martin, C. D., "A Media Taxonomy," Working Paper, Computer Science Department, George Washington University, 1995.

[11] Harada, "Anecdote : A multimedia storyboarding system with seamless authoring support," in Proceeding of ACM Multimedia, 1996.

[12] Isakowitz, T., Stohr, E. A. and Balasubramanian, P., "RMM : A Methodology for Structured Hypermedia Design," Communications of the ACM, 38(8), pp.34-44, 1995.

[13] Jeffries, R., Turner, A. A., Polson, P. G. and Atwood, M. E., "The Processes Involved in Designing Software," in J. R. Anderson, Cognitive Skills and Their Acquisition, Lawrence Erlbaum Associates, Hillsdale, NJ, pp.255-283, 1981.

[14] Kiddoo, K., "Discovering the Rules : Initiative and Teamwork in Multimedia Project Design," Multimedia Review, 3(4), pp.46-51, 1992.

[15] Kraut, R. E. and Streeter, L., "Coordination in Software Development," Communications of the ACM, 38(3), pp.69-81, 1995.

[16] Landy, J. A. and Myers, B. A., "Sketching storyboards to illustrate interface behaviors," Proceedings of CHI, pp.193-194, 1996.

[17] Lin, J., Newman, M. W., Hong, J. I. and Landay, J. A., "DENIM : Finding a Tighter Fit Between Tools and Practice for Web Site Design," Proceeding of CHI, pp.510-516, 2000.

[18] Malhotra, A., Thomas, J. C., Carroll, J. M. and Miller, L. A., "Cognitive Processes in Design," International Journal of Man-Machine Studies, 12, pp.119-140, 1980.

[19] Maybury, M. T., Intelligent Multimedia Interfaces, MIT Press, Cambridge, MA., 1993.

[20] Park, I. and Hannafin, M. J., "Empirically-Based Guidelines for the Design of Interactive Multimedia," Educational Technology Research and Development, 41(3), pp.63-85, 1993.

[21] Rosson, M. B. and Alpert, S. R., "The Cognitive Consequences of Object-Oriented Design," Human-Computer Interaction, 5, pp.345-379, 1990.

[22] Simon, H. A., "The Structure of Ill Structured Problems," Artificial Intelligence, 4, pp.181-201, 1973.

[23] Simon, H. A., The Sciences of the Artificial, 2nd ed., MIT Press, Cambridge, MA., 1981.

[24] Sonnenwald, D. H., "Contested Collaboration : A Descriptive Model of Intergroup Communication in Information System Design," Information Processing and Management, 31(6), pp.859-877, 1995.

[25] Streitz, N. A., "Designing Collaboration Environments Based on the Common Ground of Hypermedia and CSCW," Tutorial Notes of The Seventh ACM Conference on Hypertext, 1996.

[26] Thomas, J. C. and Carroll, J. M., "The Psychological Study of Design," Design Studies, July, 1(1), pp.5-11, 1979.

[27] Tway, L., Welcome to Multimedia, Management Informa-

tion Source Press, New York, NY., 1992.

[28] Walz, D. B., Elam, J. J. and Curtis, B., "Inside a Software Design Team : Knowledge Acquisition, Sharing, and Integration," Communications of the ACM, 36(10), pp.63-77, 1993.

[29] Walz, D. B., Elam, J. J., Krasner, H. and Curtis, B., "A Methodology for Studying Software Design Teams : An Investigation of Conflict Behaviors in the Requirements Definition Phases," in G. Olson, E. Soloway and S. Sheppard (eds.), Empirical Studies of Programmers, Vol.2, Albex, Norwood, NJ, pp.83-99, 1987.

백 승 익

e-mail : sbaek@hanyang.ac.kr

1990년 서강대학교 경영학과 졸업(학사)

1992년 George Washington University 경영학 석사(MBA)

1998년 George Washington University 경영학 박사(Ph.D.)

1998년~1998년 Georgia State University 객원 조교수

1998년~1999년 Saint Joseph's University 조교수

2000년~현재 한양대학교 경영학과 전임강사

관심분야 : Collaboration in Virtual Teams, CRM, Intelligent Agent Applications for E-Commerce