

WebReview: 웹 기반 논문심사 지원 시스템

서 용 무*, 전 건 호**

WebReview: Web-Based Paper Review Support System

Suh, Yongmoo, Jeon, Geonho

There has been a complaint that current editorial and refereeing system of the traditional journal systems has something to be desired due to long time lag between submission and publication, improper(unfair in some sense, because no communication is allowed between reviewers and authors) review of submitted papers sometimes, and so on. This paper introduces a Web-Based Paper Review Support System, called WebReview. The system helps to reduce the time lag and allows reviewers and authors to communicate anonymously to clarify the research results, if there is any ambiguity. WebReview is implemented in Java, using Oracle as an underlying DBMS, to capture interactions between users and the system.

* 고려대학교 경영대학

** 고려대학교 경영대학 박사학위(전공:MIS)과정

I. 서 론

전통적인 저널 시스템은 상당히 오랜 동안 명맥을 유지해 오고는 있지만, 이들의 문제점은 오래 전부터 지적되어 왔다. 몇 가지 예를 들면, 저널에는 구독자가 원치 않는 내용의 논문이 많다는 점, 논문이 게재되기까지는 너무 오랜 시간을 요한다는 점, 도서관에 비치할 경우 너무 많은 공간을 차지한다는 점, 특정 논문을 찾기가 어렵다는 점, 그리고 지면의 한계로 인해 축약된 형태로 게재를 해야한다는 점등이 있다 [Gains, 1993]. 그밖에, 때로는 논문심사자의 편견에 의해 공정한 심사를 받지 못하는 경우도 있다는 것도 하나의 단점이라고 할 수 있다. 이러한 전통적인 저널 시스템의 단점을 약간이라도 덜 수 있는 새로운 형태로 탄생하게 된 저널이 바로 전자저널, 혹은 디지털 저널이다.

처음 특정 몇몇 분야에서 실시되었던 전자저널이 점차 여러 분야로 확대 발전해 가는 추세에 있기는 하지만, 처음부터 성공적으로 운영이 되었던 것은 아니다. 컴퓨터 회의 시스템인 EIES(Electronic Information Exchange System)를 이용하여 구현하였던 한 심리학 저널은 당시의 컴퓨터 기술의 한계로 인해 사용자들의 불편을 초래해 결국 부정적인 경험으로 남게 되었고 [Hiltz, 1984], 영국에서는 논문의 내용을 효율적으로 심사하게 하는 기술의 부족뿐만 아니라 관련기관들 사이의 정치적인 알력이 전자저널의 발전을 저해하는 요인이 되기도 했다 [Okeshott, 1985]. 하지만, *The Online Journal of Current Clinical Trials*와 같은 몇몇 실험적인 전자저널의 발간은 긍정적인 평을 받기도 했다 [Gains, 1993]. 현재 인터넷 상에서 찾아보면 매우 많은 전자저널이 있음을 알 수 있지만 [Harter, 1996], 대부분은 전통적인 저널 시스템의 매체를 대체했다는 느낌을 받을 뿐, 그 단점

을 별로 제거하지는 못하고 있는 실정이다. 예를 들면, 심사 절차는 전과 같은 동일한 방법에 의존하고 있다. 참고로, [AMS, CCT, PACS, DTJ, PSY] 등이 가장 빈번하게 인용되는 전자저널 들이다.

전자저널이 성공적으로 전통적인 저널 시스템을 대체하기 위해서는, 전통적인 저널 시스템의 단점을 현재의 기술로 해결할 방법도 모색하고, 보다 근원적인 저널의 역할, 즉, 논문의 형태로 표현된 연구결과에 대한 올바른 평가, 의미 있는 연구로 평가된 연구결과의 보급 등의 역할을 어떻게 전자저널 시스템에서 지원하도록 할 것인가를 살피는 것이 반드시 필요하다.

현재의 컴퓨터 및 네트워크 기술의 발전은 전자저널의 성공적인 구현을 보장할 만한 수준에 이르고 있다. 본 논문은 컴퓨터 및 네트워크 관련 기술을 활용하여 전통적인 저널 시스템의 단점을 일부를 보완하고 (즉, 논문제출에서 게재되기까지의 시간 단축과 공정한 심사를 위한 저자와 심사위원들간의 대화 지원) 저널의 기본적인 역할 즉, 새로운 지식의 발표 및 보급을 수행하게 될 전자저널 시스템에서 필요로 하는 논문심사 과정을 지원하는 웹 기반 논문심사 지원 시스템(WebReview: Web-Based Paper Review Support System)의 개발을 목표로 한다. 이를 위해, 먼저 2장에서 개발하려는 논문심사 지원 시스템의 요구사항을 조사하고, 그에 따라 도출된 연구 문제를 기술한다. 3장에서는 관련 기술 및 연구에 대한 소개를 하고, 4장에서는 프로토타입 형태로 개발된 웹 기반 논문심사 지원 시스템의 개발 환경, 기능, 사용방법 등에 대하여 기술하고, 마지막으로 5장에서는 논문의 요약과 함께, 논문에서 소개하고 있는 웹 기반 논문심사 지원 시스템의 앞으로의 보완계획 및 기능 확장에 대하여 기술하고 끝을 맺는다.

II. WebReview 시스템의 요구사항 조사 및 문제점 도출

2.1 WebReview 시스템의 요구사항 조사

어떤 종류의 시스템이든지 개발 전에 요구사항의 조사가 필요하고 대개는 사용자 인터페이스와 기능상의 요구사항으로 대별되는 것이 보통이다. WebReview 시스템의 경우에도 이 두 가지로 요구사항을 정리해보면 다음과 같다.

먼저, 사용자들이 사용 중에 불편함을 느끼지 않아야 하고 배우기 쉬워야 한다는 전제 아래 사용자 인터페이스에 대한 요구사항은 다음과 같다.

- 1) 가능하면 기존의 웹 기반 시스템에서 사용하는 사용자 인터페이스를 따른다.
- 2) 사용 중 대부분의 입력 내용은 문자열이지만, 논문의 요약본을 표현하는데 부족함이 없어야 하므로 멀티미디어 데이터를 지원하는 web editor를 이용하여 논문을 요약하도록 한다.
- 3) 컴퓨터 화면 크기의 제약으로 인해 발생하는 불편함을 대체할 수 있는 기능이 요구된다(이는 대부분의 CSCW(Computer-Supported Cooperative Work) 분야의 응용 프로그램의 개발에서 요구되는 공통 요구사항이다[Suthers, 1995].).

다음, 기능 면에 있어서의 요구사항 정리에 앞서 전통적인 저널 시스템의 운영을 살펴보면 다음과 같다: 1) 논문모집공고(call for papers)를 한다. 2) 저자가 논문을 제출한다. 3) 편집장이 접수된 논문의 적정 심사위원을 선정하여 심사를 의뢰한다. 4) 심사위원이 심사결과를 편집장에게 보낸다. 5) 심사결과에 따라, 편집장이 저자에게 논문의 수정을 요구하거나 게재불가 또는 게재 결정을 한 후에 이를 통고한다. 6) 수정요구를 받은 저자는 논문의 내용을 수정하

여 다시 제출하고 앞의 심사과정을 다시 밟게 된다.

새로운 전자저널 시스템에서는 전통적인 저널 시스템의 기본 기능을 대부분 수용하면서 새로운 기능이 첨가되어야 하는데, 그 운영 면에 있어서 다음과 같은 점이 요구된다. 1) 논문 모집공고를 웹 상의 전자저널 홈페이지에 올린다. 2) 저자가 전자저널의 WebReview 시스템을 이용하여 논문을 제출하고, 근무처, 전자메일 주소, 전화번호 등과 같은 개인 신상에 관한 기본 정보를 입력한다. 이 때, 논문 제출 시 저자는 논문의 요약본을 시스템에서 요구하는 특정 구조에 따라 작성하여야 한다. 3) 편집장은 심사위원을 선정하고, 각 심사위원에게 심사의뢰를 전자메일을 통하여 한다. 4) 심사위원은 전자저널의 홈페이지에서 심사의뢰를 받은 논문을 찾아가서 심사위원의 개인 정보를 입력하고, 해당 논문의 요약본을 본 후 (요약본에 불명확한 점이 있으면 원본을 본 후) 시스템에 정의된 구조에 따라서 자신의 심사의견을 제시한다. (심사의견을 제시한 후 심사위원이 세션을 끝내면, 시스템은 저자에게 전자메일을 통해 심사의견이 있음을 통보한다.) 5) 저자는 심사의견에 대해 방어할 필요가 있을 때에는 시스템에서 허락하는 구조로 다시 자신의 의견을 덧붙인다. (저자가 세션을 끝내면 역시 시스템은 해당 심사위원에게 전자메일을 통해 저자가 심사의견에 대해 답을 했음을 통보한다.) 6) 앞단계 4, 5를 통해 심사위원과 저자간에 대화를 반복하면서 일정기간이 지나면 편집장이 심사위원들에게 전체 대화 내용을 참조하면서 최종 심사의견을 제시할 것을 요구한다. 7) 편집장은 심사위원들의 심사의견을 종합하여, 수정을 요구하거나, 게재 불가 혹은 게재 가의 결정을 하여 저자에게 통보를 한다. 8) 만일 수정요구가 있을 때는 저자는 수정사항에 대하여 응답을 하고, 시스템에서는 편집위원장에게 전자 메일을 통해서 통보를 한다. 9) 편집위원장(편집위원회)

이 게재여부에 대한 최종 결론을 내린다.

이상에서의 내용을 정리하면 WebReview 시스템의 기능 면에서 요구되는 점은 다음과 같다:

- 1) 논문의 심사위원 선정을 도울 수 있는 심사위원군이 데이터베이스화되어야 한다. 논문의 내용에 적합한 심사위원이 선정될 수 있도록 심사위원군에 있는 심사위원들의 전공분야, 관심분야 등 필요한 정보를 데이터베이스에 저장하여 활용을 하도록 한다.
- 2) 논문의 내용 중에서 심사에 필요한 내용을 요약하여 심사과정에서 활용하도록 한다. 논문의 연구문제, 연구방법 및 공헌 등을 명확히 표현하는데 필요한 기본 구조를 시스템에서 제공하여야 한다. (만일 더 자세한 내용이 필요할 때만 논문의 원본을 참고하도록 한다.)
- 3) 필요시, 논문의 저자와 심사위원과의 의견 교환 즉 대화를 통하여 논문에 대한 공정한 심사가 이루어지도록 한다. 저자가 작성한 논문의 요약을 심사위원이 보았을 때, 만일 불분명한 부분이 있거나 애매한 점이 있을 때에는 연구에 대한 공정한 심사를 위해서 저자와 심사위원간에 대화를 할 수 있는 기회가 주어져야 한다.
- 4) 전통적인 저널 시스템에서의 단점인 긴 심사기간을 단축하기 위해서 저자와 심사위원들의 대화가 신속하게 이루어지도록 하여야 한다.

2.2 WebReview 시스템의 구현을 위한 연구과제 도출

이상에서 살펴본 바와 같이 전자저널을 구현하기 위해서는 다음과 같은 문제들이 우선 해결되어야 한다.

첫째, 논문의 저자는 논문의 제출과 함께, 논

문의 요약본을 제출하여야 하는데, 요약본을 통하여 저자는 연구결과에 대하여 함축적이면서도 연구의 요지를 분명하게 나타낼 수 있어야 한다. 따라서, 어떤 형태 또는 구조로 요약을 하는 것이 바람직한지 이에 대한 연구가 필요하다. 논문의 내용을 요약하여 구조적으로 보기 쉽게 하여 준다면 논문을 심사함에 있어 많은 시간이 절약 될 수 있고, 또한 심사위원의 의견을 논문 요약과 함께 표현할 수 있으며, 저자는 심사위원의 각 의견에 대해 자신의 의견을 피력할 수도 있다. 이 경우, 어느 부분이 심사위원의 의견인지 그리고 어느 부분이 저자의 표현인지가 구분이 가는 구조이어야 한다. 물론 연구의 분야 및 방법에 따라서 연구 내용을 표현하기 위한 구조가 다를 수 있음을 염두에 두어야 한다.

둘째, 요약본의 내용을 본 후 심사위원이 심사의견을 덧붙이게 되면 심사위원과 저자간의 대화를 통하여 연구내용의 중요성을 파악하게 된다. 따라서, 심사위원과 저자간의 대화를 효과적으로 지원하기 위한 일종의 대화 규칙이 필요하게 된다. 공정한 심사를 위해서는, 저자의 논문 요약이나 심사자의 심사 의견에 대하여, 저자, 심사위원 쌍방간에 의견 교환의 형식으로 토론이 가능해야 한다. 예를 들면, 저자의 논문 요약에 대하여는 어떤 심사위원도 심사 의견을 표할 수 있으며, 한 심사자가 논문 요약(내용)에 대하여 질문을 하였다면, 저자가 이에 대하여 답변을 할 수 있어야 하며, 이어서, 답변에 대하여도 새 질문을 하는 등의 공격을 할 수도 있다.

셋째, 심사위원들이 논문의 내용 요약을 보고 또는 요약에 대한 의견 교환을 통하여 파악하게 되면, 최종적으로 기존의 저널 시스템에서와 같은 또는 비슷한 방식으로 논문의 개별 평가를 할 수 있어야 한다. 따라서, 시스템에서는 심사위원 개인이 평가를 하기 위한 방법이 있어야 하고, 동시에 편집장이 최종 평가를 하기

위한 지원을 해야 한다.

넷째, 이러한 세부 연구과제 외에도, 사용자 인터페이스에 관한 연구와 사용자의편의를 위한 다양한 서비스를 고려하여야 한다.

III. WebReview 시스템의 구현에 영향을 준 연구에 대한 조사

3.1 IBIS(Issue-Based Information Systems)

IBIS 모델은 명확한 답이 없거나, 쉽게 풀리지 않는 복잡한 문제에 대하여 여러 사람이 토론을 거쳐서 그 해결책을 찾으려고 하는 경우를 지원하기 위하여 개발된 모델이다[Rittel, 1973]. 이러한 문제의 특성은 아무도 이러한 문제들이 옳은 해답을 가지고 있다고 단언할 수 없음을 내포한다. 그러므로 이러한 문제에 대하여는 여러 가지의 상반된 또는 비슷한 해답이 거론 될 수도 있으며, 따라서, 거론된 각각의 해답에 대하여 토론자들의 충분한 의견 교환을 필요로 한다. 그리고, 이러한 복잡한 문제에 대한 토론은 한번의 토론에서 원하는 해답을 찾아내기란 어렵기 때문에 여러 번 계속적으로 토론을 해야 하고 따라서 매번 토론의 과정에서 도출된 내용을 보관하고 다음의 토론으로

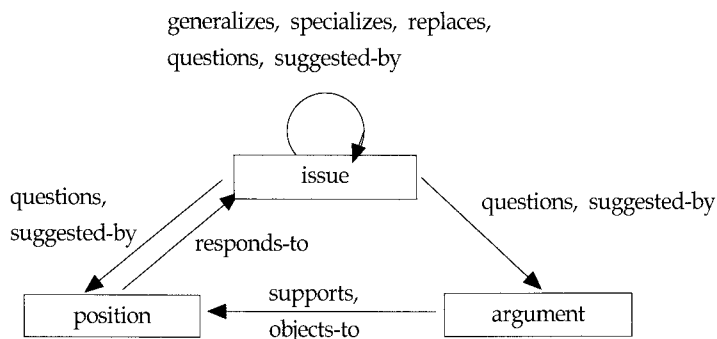
이어지게 하여야 한다.

IBIS 모델의 근본 원리는 쉽게 풀려지지 않는 복잡한 문제에 대한 해답은 그럴듯한 해답들의 큰 창고가 만들어 진 후에 얻어 질 수 있다는 것이다. IBIS 모델은 이런 해답 창고를 개발하는 노력에 시스템적, 구조적 접근을 하게 한다. IBIS 모델은 토론 이론에 기초한다. 이 모델 내에서 토론 과정은 여러 가지 노드와 링크의 네트워크로 표현된다. 기본 노드 타입은 ISSUE, POSITION 및 ARGUMENT의 세 타입으로 이루어진다. 간단히 설명하면, ISSUE 노드는 질문의 형식으로 된 복잡한 문제를 의미하고, POSITION 노드는 한 문제(issue)에 대한 제안된 해답을 표시하며 ARGUMENT 노드는 제안된 해답(position)을 찬성하거나 또는 반대하는 문장이나 평가를 나타낸다. 링크(relationship)는 노드 사이의 관계를 정의한다. IBIS 모델의 노드 타입과 링크를 그림으로 표시하면 <그림 3-1>과 같다.

IBIS 모델은 여러 종류의 토론을 지원하기 위해서 개발된 많은 시스템에서 참조되었다 [Conklin, 1988; Rein, 1990; Pott, 1987; McCall, 1987].

3.2 Discussion Rules

토론의 효과적인 진행을 위해서는 토론을 통



<그림 3-1> IBIS 모델의 기본 구조

제하기 위한 규칙이 필요하게 된다. Delinger [1972]은 이러한 규칙을 알아내기 위해서 토론 중 가능한 다섯 가지의 서로 다른 move를 정의하고, 토론 중 어떤 사람이 제시한 의견에 대하여 다른 사람이 어떤 종류의 move를 할 수 있는지를 체계적으로 조사하였다. 즉, 토론자들이 어떤 의견에 대하여 토론을 한다는 것은 의견에 대한 challenge와 이에 대한 response의 연속으로 생각하고, challenge를 question, contest, rebut, defend, concede의 다섯 가지로 구분하였다. 각각의 challenge에 대하여 가능한 response로 다시 이들 다섯 가지 중의 하나가 가능하다고 보았다. 여기서, question은 제안된 의견의 타당성에 대한 질문이고, contest는 의견의 내용과 모순이 되는 새로운 의견을 제안하는 경합을 의미하고, rebut는 제안된 의견의 비 타당성을 보이는 반박을 의미하며, defend란 의견의 정당성을 보이기 위한 증거제시를 말하며, 마지막으로 concede는 challenge에 대하여 아무런 response를 하지 않고 받아들임을 의미한다. 토론에서는, 이들 다섯 가지의 challenge 각각이 다시 다섯 가지의 response 중의 하나로 이어지는 challenge와 response가 반복된다. 그러나 모든 challenge와 response 짝을 다 허용하지 않고, 성공적인 토론을 위해서 필요한 8개의 규칙만을 고려함으로써, 상당수의 challenge와 response 짝을 제거하였다. 예를 들면, 질문이 질문으로 이어지는 토론은 의미가 없기 때문에 질문 다음에 질문은 허용하지 않는 것이 당연하다.

3.3 기존 관련연구 결과의 활용

<그림 3-1>의 IBIS 구조에 의하면, 상당히 복잡한 형태의 그래프로 토론의 내용을 나타낼 수가 있다. 예를 들면, ISSUE 노드들 사이에, 한 ISSUE에서 새로운 ISSUE가 도출되어 이들 사이에 generalizes/specializes/replaces/questions 등의 관계가 성립하게 되는 경우도 가능하고, ARGUMENT 노드와 ISSUE 노드 사이에도

questions/suggested-by의 관계가 성립될 수도 있다. 하지만, 이러한 세 가지 노드들 사이의 복잡한 관계를 컴퓨터 시스템에서 지원을 하게 되면, 화면 위에 이를 그래프 형태로 표시할 경우 너무 복잡하게 되어 오히려 사용상의 장애 요인이 될 수도 있다.

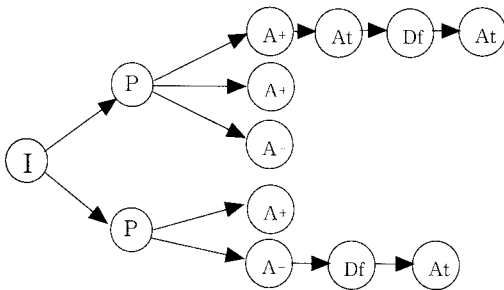
따라서, 본 논문에서 개발하고자 하는 논문심사 지원 시스템인 WebReview 시스템에서는 가장 간단하고 가장 기본적인 관계만을 나타내도록 IBIS모델을 단순화하여 구현하였다. 논문의 저자는 자신의 연구 결과를 단순화된 IBIS 모델로 다음과 같이 표현하게 된다. 즉, 논문에서 해결하고자 하는 연구 문제를 ISSUE로, 연구 문제에 대한 저자의 접근 방법을 하나의 POSITION으로, 동시에 같은 연구 문제에 대하여 연구 과정에서 조사한 기존 연구에서의 접근 방법을 또 다른 POSITION으로, 그리고, 각각의 접근 방법인 POSITION에는 이를 뒷받침하거나 반대하는 내용의 ARGUMENT가 생성된다. (그러나, 연구문제에 대한 접근방법을 하나의 POSITION으로 나타내기에는 여러 가지로 부족한 점이 있어서, 앞으로 WebReview 시스템을 확장할 때에는 연구내용에 관련된 다양한 내용을 표현하기에 적합한 구조에 대한 연구가 필요하다.)

논문 심사를 위한 과정에서도 논문의 내용에 대하여 저자와 심사사간에 토론을 거치게 하는 것이 공정한 심사를 위해서는 필요하다고 본다. 그렇다면, 이 과정을 컴퓨터 시스템을 통하여 거치도록 하기 위해서는 어떻게 구현해야 하는가를 고려해야 한다. 우선, 3.2에서 소개한 연구의 내용을 그대로 구현할 수도 있지만 컴퓨터 시스템으로 구현하기에는 다소 복잡하다는 생각이므로, 이를 구현하기 좋게 단순화 할 필요가 있어서, 주장한 내용(주로, ARGUMENT)에 대하여 단순히 심사위원의 attack과 저자의 defense를 반복하는 것으로 하였다.

따라서 WebReview를 이용한 저자와 심사위원간의 토론은 다음과 같이 진행된다. 저자가 연구 결과를 ISSUE-POSITION-ARGUMENT의

형식으로 표현해 놓으면, 심사위원들은 각각 그 내용을 살피고, 반대되는 새로운 POSITION을 제시하면서 자신의 POSITION을 뒷받침하는 ARGUMENT를 첨가하거나, 저자의 POSITION에 반대하는 ARGUMENT를 첨가할 수도 있다. 이어서, 저자의 POSITION을 뒷받침하는 ARGUMENT를 attack할 수도 있다. 저자의 POSITION에 반대하는 내용의 ARGUMENT를 첨가하거나 저자의 POSITION을 뒷받침하는 ARGUMENT를 attack하는 경우에는 저자가 defense를 해야 한다. 저자가 defense를 했는데 만족스럽지 못할 경우에는 심사자가 다시 defense 내용에 대하여 attack을 할 수도 있다.

<그림 3-2>는 토론 중의 내용을 나무구조 형식으로 표현한 것이다. 한 ISSUE(I)노드에 대해 두 개의 POSITION(P) 노드가 있으며 각 POSITION(P) 노드에 대해 positive ARGUMENT(A+) 노드와 negative ARGUMENT(A-) 노드가 표현되어 있다. 그리고 검증 단계에서 토론된 attack(At) 노드와 defense(Df) 노드가 첨가되어 있다.



<그림 2-2> 토론의 진행을 나타내는 나무구조

<표 4-1> 컴퓨터 환경

	server	client
Hardware	586 PC (100MHz) main memory: 32MB	486 PC 이상
Software	OS: windows 95 DBMS: Personal Oracle 7.3 Webserver: JavaWebServer 1.1 beta Compiler: JDK 1.1.4	WebBrowser: Netscape Communicator 4.03 + Java Patch 1.1 Library: swing.jar, rose.jar, classes111.zip

요약하면, WebReview의 구현에는 기존의 두 연구의 결과를 접합하여, 하나는 전자저널에 제출된 논문의 요약에, 다른 하나는 요약된 논문의 내용에 대하여 심사위원과 저자사이에 대화를 통하여 논문의 내용을 검증하는데 사용하도록 하고 있다.

IV. WebReview(Web-Based Paper Review Support System)

이 장에서는 앞에서 언급한 문제점들을 해결하고 인터넷 상에서 논문을 관리하고 심사할 수 있는 시스템의 설계와 구현에 대하여 설명하고자 한다. WebReview 시스템은 인터넷을 통한 Client/Server 구조로 되어 있다. 따라서 본 시스템 사용자는 인터넷이 연결된 어느 곳이나 웹 브라우저를 통하여 사용할 수 있다.

4.1 개발 환경

client/server 구조

먼저 웹서버 환경에 대하여 설명하면, 586 PC (100MHz, 32MB)에 windows 95하에서 DBMS로는 Personal Oracle 7.3을 사용하였고, JavaWebServer 1.1 Alpha를 웹서버로, JDK 1.1.4 compiler를 사용하였다. Client 환경으로는 486급 PC이상에서 web browser로 Netscape Communicator 4.03, 필요한 Patch File로는 NETSCAPE COMMUNICATOR 4.0 JDK 1.1 SUPPORT, 그리고 swing.jar, rose.jar, classes111.zip

라이브러리 file이 필요하다. <표 4-1>에 개발 환경을 요약하였다.

Java

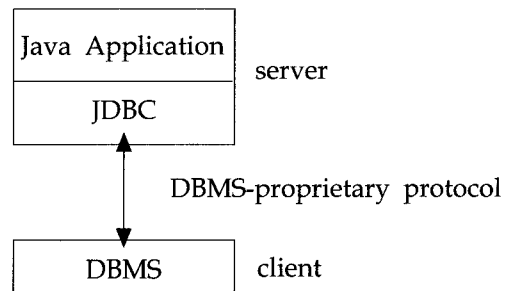
프로그래밍 언어로는 Java를 택하였는데, 그 이유로는 다음과 같이 여러 가지가 들 수 있다. 먼저, Java는 다른 언어에 비해 코딩이 간단하며, 임의의 시스템 자원을 접근하는 것이 불가능하기 때문에 안전하고, 단일 상속만을 지원하는 객체지향적 언어이며, garbage collection 및 exception handling 기능이 있어서 메모리 관리나 예외상황처리에 강하다고 할 수 있다. 한편, Java는 기계 중립적인 bytecode로 실행 화일이 생성되므로 어느 컴퓨터에서나 쓰일 수 있다는 점 때문에 환영받고 있는 것이다. 그리고, Java는 기계 중립적인 bytecode로 미리 컴파일 된 것을 실행할 기계어 코드로 번역하여 수행하는 방식을 채택하고 있고 또 bytecode는 기계어로 번역될 것을 감안하여 설계되었으므로 그 수행 능력이 C나 C++ 컴파일러에 의한 기계어 코드에 견줄만하다. 특히 Java는 멀티 프로세스 하드웨어를 지원하도록 설계되었으므로 멀티 CPU 시스템에서 높은 효율을 낼 수 있으며 멀티 스레딩을 지원하고 있다. 무엇보다 중요한 것은 Java는 미리 필요로 되는 기능들을 제공함으로써 쉽게 프로그래밍을 할 수 있게 한다는 것이다. 인터넷 환경을 위하여 TCP/IP networking, WWW and HTML, Distributed Program 클래스들을 제공함으로써 쉽게 인터넷 프로그램을 할 수 있으며 기본 코어 클래스로서 Language, Utilities, Input/Output, Low-Level Networking, Abstract Graphical User Interface Class들을 제공한다.

JDBC

JDBC는 Java API이다. 그렇지만 종종 자바 데이터베이스 연결기(Java Database Connectivity)라고 생각되어 진다. JDBC는 자바 프로그램으로 만들어진 라이브러리(클래스와 인터페이스)

의 집합으로 이루어져 있다. JDBC는 틀이나 데이터베이스 개발자들을 위한 표준 API를 제공하고 순수 자바 API로 데이터베이스 애플리케이션을 만들 수 있게 한다. 기본적으로 데이터베이스와의 연결을 해주는 기능을 한다. 마이크로 소프트사의 ODBC(Open DataBase Connectivity) API는 관계형 데이터베이스를 연결하는데 폭 넓게 쓰이고 있다. 또한 거의 모든 플랫폼에서 거의 모든 데이터베이스와 연결할 수 있다. 하지만 자바에서는 바로 쓸 수는 없으므로 JDBC-ODBC Bridge의 형태로 쓰인다.

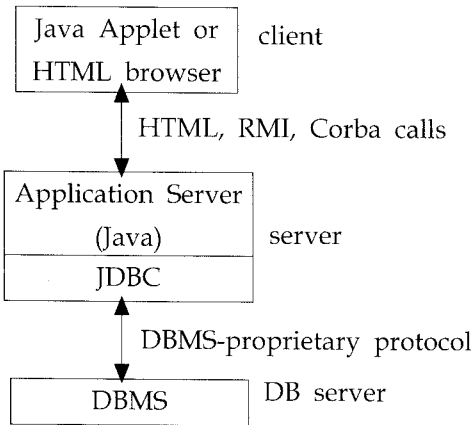
JDBC는 two-tier model과 three-tier model을 모두 지원한다. 전자에서는 Java applet이나 애플리케이션이 데이터베이스와 직접 연결되는데 DBMS와 통신할 수 있는 JDBC 드라이버가 필요하다. 네트워크를 통해 사용자의 SQL문이 local 이나 remote 지역에 있는 DBMS 시스템으로 보내지고 결과 값을 받는다. <그림 4-1>은 two-tier 모델을 나타내고 있다.



<그림 4-1> JDBC Two-tier 모델

후자의 경우에는, 먼저 command가 중간 계층(middle tier)으로 보내지면 그 곳에서 SQL문이 데이터베이스관리 시스템으로 보내어 진다. 데이터베이스 관리 시스템은 그 SQL문을 처리해서 결과 값을 중간 계층으로 보내면 다시 그것을 사용자에게 보내준다. 이것은 데이터베이스로의 접근 및 데이터의 통제가 가능하고 쓰기 편한 higher-level API를 사용할 수 있으며 많은 경우에 있어서 성능상의 장점을 가질 수

있다. 성능때문에 중간 계층이 C나 C++로 많이 쓰여졌지만 속도의 향상과 Java의 장점을 가질 수 있기 위해서는 Java 중간 계층이 필수적이다. <그림 4-2>은 three-tier 모델을 설명하고 있다.



<그림 4-2> JDBC Three-tier 모델

JDBC의 종류에는 다음의 네 가지가 있다. 이 중, Category 3과 4가 internet과 intranet 환경에서 자동으로 설치(예를 들면, applet과 함께 JDBC driver를 함께 download 하는 것)가 가능하기 때문에 즉, Java의 장점을 제공하기 때문에 사용이 권장된다. 참고로, WebReview는 Type 4로 구현이 되었지만, 다른 타입으로 쉽게 변경이 가능하다.

1) Type 1: JDBC-ODBC bridge + ODBC driver + DB

Type 1 driver는 JDBC가 ODBC driver를 통해서 DB를 접근하게 한다. 따라서 각 client에 ODBC binary code가 설치되어야 하며 Three-tier architecture에서 Java로 쓰여진 application server code에 적당하다.

2) Type 2: Native-API + partly-Java driver

Type 2 driver는 JDBC call을 각 DBMS를 위해 client API에 맞는 call로 바꾸어 준다. 따라서 Type 1 driver처럼 각 client에 binary 코드가 설치되어야 한다.

3) Type 3: JDBC-Net + pure Java driver

이 driver는 JDBC call을 DBMS-independent net protocol로 변환한다. 서버에서 다시 이 protocol이 DBMS protocol로 변환된다. 이 net server middle-ware는 많은 다른 종류의 데이터베이스에 순수 Java client가 연결될 수 있게 한다.

4) Type 4: Native-protocol + pure Java driver

이 driver는 JDBC call을 DBMS에 곧 바로 사용 될 protocol로 변환한다. 이것은 client가 DBMS server로 직접 call을 보낼 수 있으며 intranet을 위한 실질적인 솔루션이다.

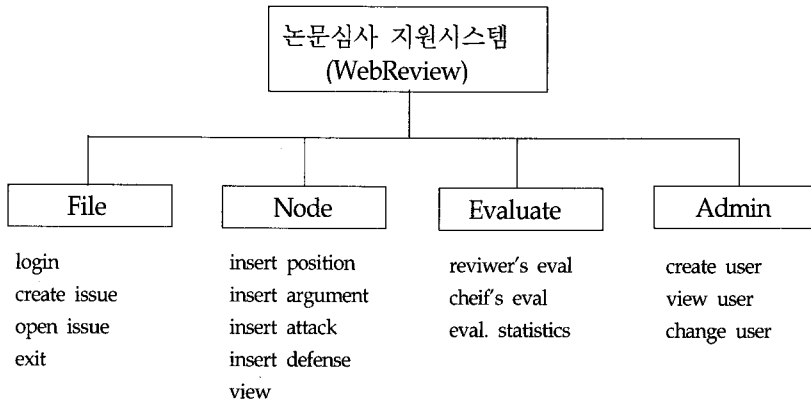
4.2 WebReview 설계

프로그램의 구성

프로그램 구성도는 <그림 4-3>에서 보는 바와 같다. 처음 시스템에 login하고 issue의 생성을 제공하는 File 메뉴, IBIS의 구조에 맞게 노드를 만들고 검증 단계의 의견을 올릴 수 있는 Node 메뉴, 심사위원들의 평가를 입력하고 조회해 볼 수 있는 Evaluate 메뉴 그리고 사용자 관리를 위한 Admin 메뉴가 있다. 각 메뉴에서 할 수 있는 세부 기능을 살펴보면 다음과 같다.

1) File 메뉴

다른 사용자로 새로 login 하거나 논문요약의 생성 또는 논문을 보기 위해 선택하는 메뉴로 이루어져 있다.



<그림 4-3> WebReview 구성도

- Login: 시스템을 시작하기 위한 login 메뉴
- Create Issue: Issue의 생성을 위한 메뉴
- Open Issue: 트리 구조로 보기 위해 Issue를 여는 메뉴
- Exit: application 모드로 실행될 때 시스템을 종료한다.

2) Node 메뉴

논문의 제출자가 논문의 내용을 IBIS 형식으로 입력할 수 있는 메뉴가 제공된다. 나무 구조 화면에서 특정 issue나 노드를 선택하면 그 노드에서 추가할 수 있는 노드에 대한 메뉴가 자동으로 선택 가능하게 되며 추가 할 수 없는 메뉴는 선택 할 수 없게 된다. 또한 각 노드에 대한 입력자나 입력 시간, 그리고 자세한 내용을 볼 수 있는 메뉴도 제공 된다.

- Insert Position: issue 노드에 대한 position 노드를 생성하는 메뉴 (issue 나무 구조 화면에서 issue가 select 되어야 실행이 가능)
- Insert Argument(+): position 노드에 대한 supporting argument 노드를 생성하는 메뉴 (issue 나무 구조 화면에서 position이 select 되어야 실행이 가능)

- Insert Argument(-): position 노드에 대한 object argument 노드를 생성하는 메뉴 (issue 나무 구조 화면에서 position이 select 되어야 실행이 가능)
- Insert Attack: argument 노드에 대한 attack 노드를 생성하는 메뉴 (issue 나무 구조 화면에서 positive argument 또는 defense가 선택되었을 때 실행이 가능)
- Insert Defense: attack 노드에 대한 defense 노드를 생성하는 메뉴 (issue 나무 구조 화면에서 negative argument 또는 attack 노드가 선택되었을 때 실행 가능)
- View Node: 각 노드의 내용을 보기 위한 화면으로 바꾸는 메뉴

3) Evaluate 메뉴

심사자의 권한과 평가자의 권한을 가진 사용자만이 사용 가능하며 한 issue가 선택되어 졌을 때 사용 권한에 맞게 가능한 메뉴가 자동적으로 선택 가능하게 된다.

- Reviewers Evaluation: 심사위원의 권한을 가진 사용자가 login 하면 나타나는 메뉴로 특정 issue를 open 했을 때 사용 가능하며 다섯 가지 평가 항목을 1점에서 5점까지 점수를 줄 수 있게 된다.

- **Chiefs Evaluation:** 편집위원장의 권한을 가진 사용자가 login 해야 사용할 수 있는 메뉴로 논문의 내용과 토론 내용 및 심사위원들의 점수를 토대로 논문을 4단계 중 한 개의 상태로 평가하게 된다.
- **Eval Statistics:** 심사위원의 권한으로 사용할 수 있으며 심사위원들의 심사 count, 심사한 내용의 항목별 평균 점수 및 표준 편차 값을 조회 할 수 있다.

4) Admin 메뉴

시스템 관리자의 권한으로 login했을 때 사용할 수 있으며 시스템 사용자에게 대한 리스트 보기, 생성 및 권한 부여 등의 관리 업무를 사용할 수 있다.

- **Create User:** 사용자를 생성하는 메뉴
- **View/Change User:** 각 사용자에게 대한 자세한 정보를 보고, login id를 제외한 다른 정보의 수정을 할 수 있다.

사용자 권한

<표 4-2>에서 보듯이, 본 시스템의 사용자는 일반 사용자(논문 저자 포함), 심사위원 및 관리자의 세 부문으로 나뉘어 지며 각 사용자에게 대한 권한을 부여하여 권한에 맞는 작업을 할 수 있게 하였다. 먼저 논문 저자는 IBIS 모델 형태로 논문을 upload 한다. 다음으로 인터넷상에서 토론을 통한 검증 단계를 거친 후 심사위원들이 심사를 하고 최종 평가를 받게 된다. 시스템 관리자는 이 시스템의 사용자들을 관리하게 된다.

<표 4-2> 사용자 권한

	논문 저자	심사위원, 편집위원장	관리자
논문 제출	○		
논문요약본 작성	○		
토론 참여	○	○	
논문 평가		○	
사용자 관리			○

데이터베이스 테이블

본 시스템 구현에 필요한 DB 테이블은 IBIS 모델에 따른 논문 정보와 검증 단계에서 발생하는 토론 정보를 저장하는 NodeTable, 각 논문의 심사 정보를 저장하는 EvaluateTable, 그리고 사용자 관리를 위한 UserTable로 이루어져 있다.

1) NodeTable(nodeId,parentId,userId,subject,nodeType,iDate,eStatus,content)은 논문의 저자나 토론 참가자가 WebReview 시스템을 사용하면서 입력하는 정보를 저장하며 각 필드는 다음과 같다.

- **nodeId:** 노드의 고유 식별자(새로 생성되는 노드마다 고유 식별자가 부여된다.)
- **parentId:** 노드의 부모노드의 식별자(issue 노드일 때는 0의 값을 가짐)
- **userId:** 노드를 입력한 사용자의 식별자
- **subject:** 내용을 대표하는 문장
- **nodeType:** 노트타입
- **iDate:** 입력한 날짜 및 시간
- **eStatus:** 평가 내용(e.g., accept, reject 등을 나타내는 코드)
- **content:** 노드의 내용

2) EvaluateTable(userId,issueId,iDate,field1,...,field5)은 심사위원들의 심사내용과 관련된 정보를 저장하며 각 필드는 다음과 같다.

- **userId:** 심사위원에게 할당된 식별자
- **issueId:** issue 노드의 식별자
- **iDate:** 심사 일 및 시간
- **field1 - field5:** 각 심사 항목별 점수

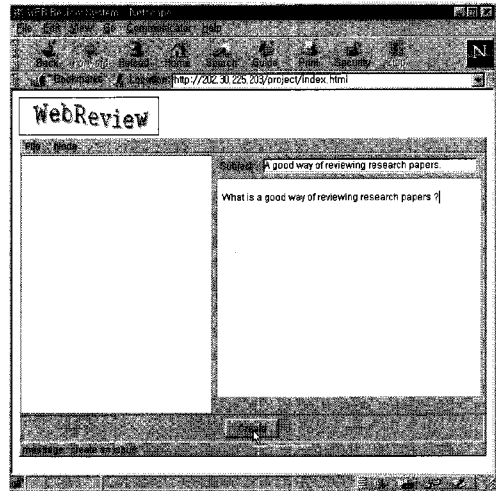
3) UserTable(userId,login,password,name,eMailAddr,phone,priority,resArea)은 시스템 사용자의 정보를 저장하며 각 필드는 다음과 같다.

- **userId:** 사용자의 고유 식별자
- **login:** login id

- password: 사용자의 암호
- name: 사용자의 이름
- eMailAddr: 사용자의 전자메일 주소
- phone: 사용자의 전화번호
- priority: 사용자 권한을 나타내는 등급
- resArea: 사용자의 연구분야

WebReview 사용 예

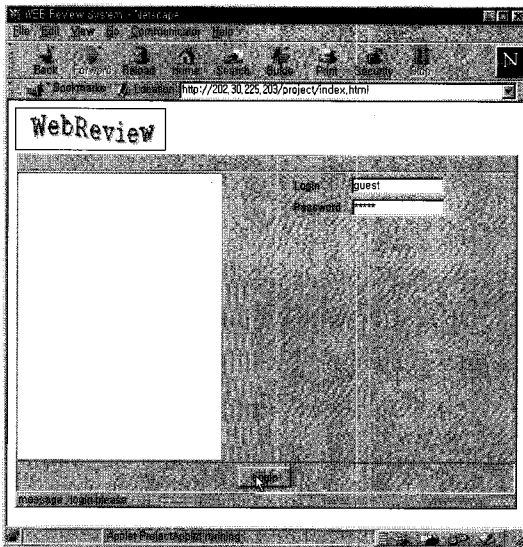
여기서는 WebReview 세션을 보여주는 화면에 대한 소개를 하려고 한다. 먼저, <그림 4-4>는 처음에 논문의 저자, 심사위원 등의 사용자가 WebReview 시스템에 login하는 화면이다. login한 후, File메뉴에서 create issue를 선택하면 생기는 <그림 4-5>는 논문의 저자가 요약본을 작성할 때 다루고자하는 문제를 issue로 등록하는 화면이다.



<그림 4-5> Issue Creation

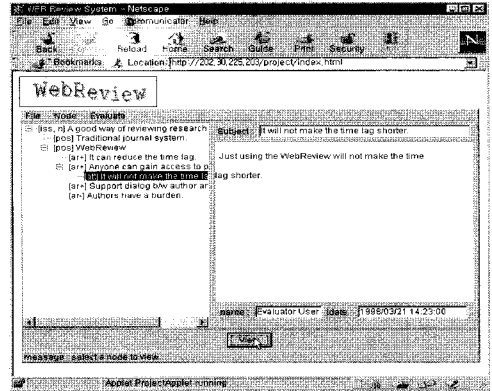
issue를 등록한 후 저자는 먼저 등록된 여러 개의 issue 중에서 <그림 4-6>에서와 같이 먼저 해당 issue를 고르고 이어서 논문에서의 연구결과를 position과 argument로 입력을 하여야 한다. <그림 4-6>에서 issue-list 중에 보면 [iss,n], [iss,a] 등이 있는데 전자는 해당 issue가 아직 심사가 되지 않은 상태를, 후자는 심사에서 accept된 상태를 나타낸다. <그림 4-7>은 한 논문의 연구내용에 대한 요약본 작성을 마친 상태를 보여주고 있다. ISSUE 노트 하나에 2개의 POSITION이 있으며, 2번째 POSITION에는 3개의 supporting argument가 입력되어 있음을 나타낸다. 이렇게 저자가 WebReview 세션을 마치게 되면 WebReview 시스템은 자동으로 심사위원들에게 전자 메일을 통해서 심사의뢰를 하게 된다(구현중임). 각 심사위원은 <그림 4-5>에서와 같이 WebReview 시스템에 login한 후 <그림 4-8>에서 보는 바와 같이 저자가 주장한 여러 내용에 대하여 자신의 의견을 제시할 수 있다.

현재 <그림 4-8>에서는 WebReview 시스템을 쓰는 것만으로는 논문심사 과정에서의 시간을 단축할 수 없다는 의견을 제시하고 있는데, 저자는 이에 대해 <그림 4-9>에서와

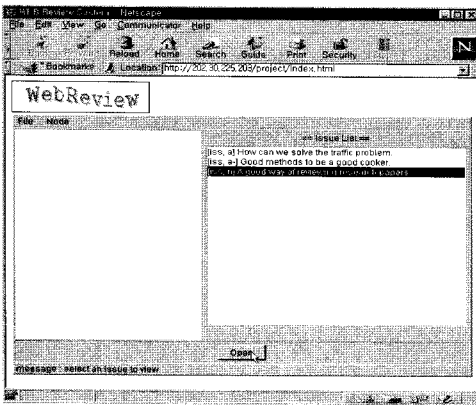


<그림 4-4> WebReview login

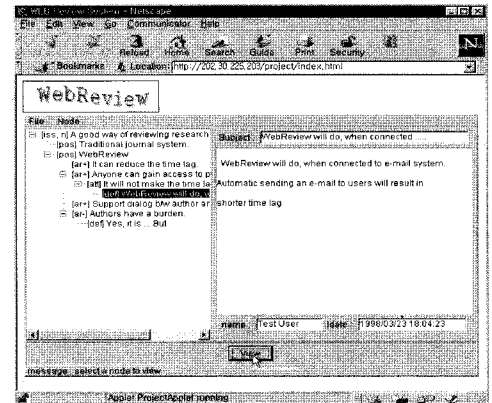
같이 전자메일 시스템을 연결하여 각 사용자에게 심사과정에서 해야 할 일이 있다는 통고를 하게 되면 심사시간의 단축이 가능하다는 응답을 하고 있다. 이와 같이 저자와 심사위원들이 논문의 내용에 대하여 대화하듯이 검증하는 단계를 갖게 되므로 보다 공정한 심사를 기할 수 있다. 저자와 심사위원들간의 대화가 일정기간동안 계속된 다음 WebReview 시스템에서는 각 심사위원들에게 최종 심사 의견을 제시하라고 전자메일을 통하여 알려주게 되면 각 심사위원들은 <그림 4-10>, <그림 4-11>에서와 같이 각 평가항목에 대하여 평가를 하고 논문에 대한 최종 평가를 하게 된다.



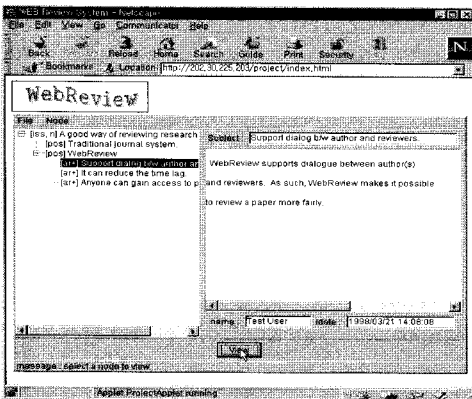
<그림 4-8> 심사위원의 논문 내용에 대한 의견 제시



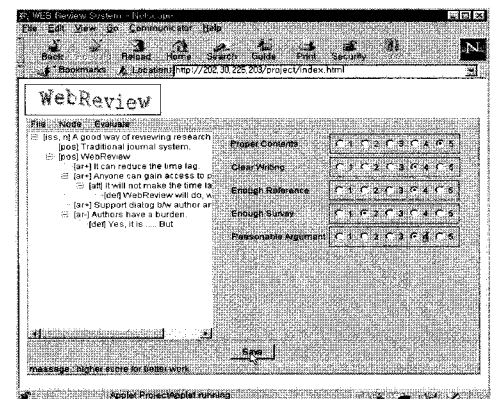
<그림 4-6> issue 고르기



<그림 4-9> 심사의견에 대한 논문 저자의 설명

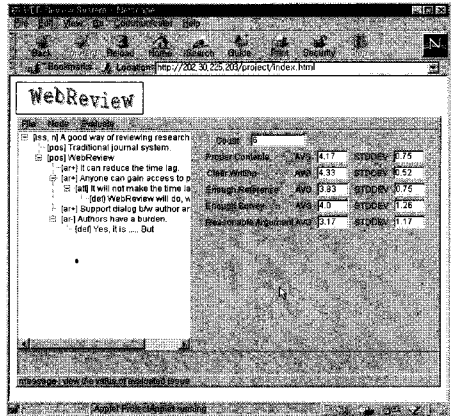


<그림 4-7> 저자의 논문 요약본 작성

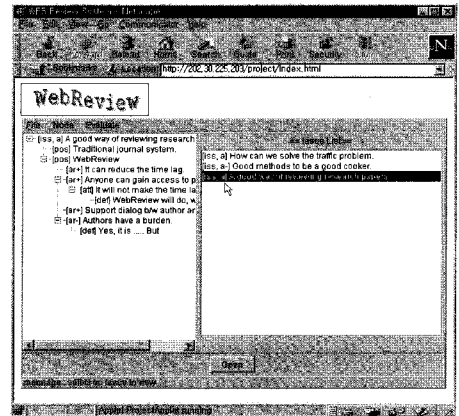


<그림 4-10> 각 심사위원의 항목별 평가

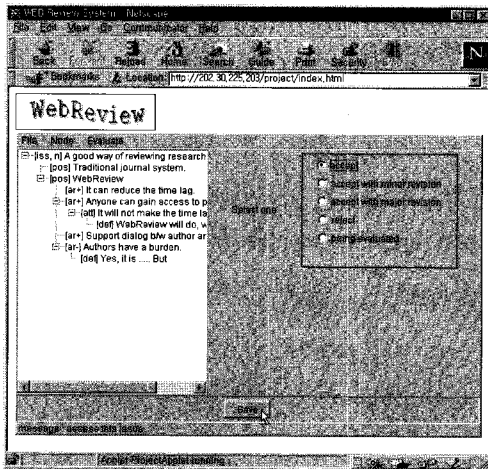
이 후, 편집위원장은 <그림 4-11>과 <그림 4-12>에 보듯이 심사위원들의 개인별 최종평가와 평가항목에 대한 평균점을 참조한 후에 <그림 4-13>에서와 같이 논문의 게재여부를 결정하게 된다. <그림 4-13>에서 화살표가 가리키는 곳을 보면 [iss,n] 이었던 것이 [iss,a]로 바뀌어졌음을 알 수있다. [iss,n]은 해당 issue의 논문이 아직 최종 평가를 받지 않았다는 것을 의미하고, [iss,a]는 해당 issue의 논문이 accept되었다는 것을 나타낸다. 물론, 수정 후 재심사 등의 필요가 있을 때에는 저자는 수정사항에 대하여 응답을 하고, 시스템은 이를 편집위원장에게 통보한다. 이때, 수정사항에 대한 응답시, 논문의 수정 부분을 참조할 수 있도록 이를 알려준다. <그림 4-14>는 심사위원들의 논문에 수정사항을 제시한 것에 대하여 저자가 이를 어떻게 수정에 응답을 하고 있는지를 보여주고 있다.



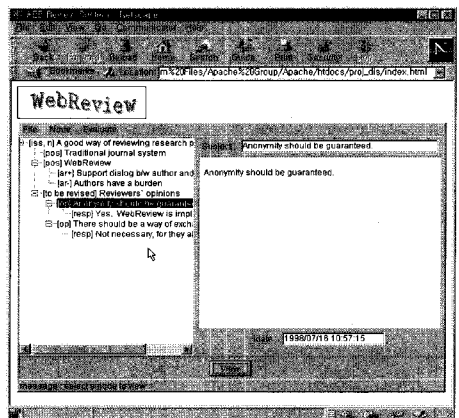
<그림 4-12> 심사위원들의 항목별 평가의 통계



<그림 4-13> 편집위원장의 최종평가



<그림 4-11> 각 심사위원의 최종 평가



<그림 4-14> 심사위원들의 요구사항 입력

V. 결 론

본 연구에서는 전통적인 저널 시스템의 논문 심사과정의 개선에 초점을 두고 시스템을 설계하고 최근 네트워크 개발 언어로 떠오르고 있는 Java 언어를 사용하여 개발하였다. 따라서 인터넷 환경에서 실제로 쉽게 적용할 수 있으며 구현된 WebReview 시스템의 특징은 다음과 같다.

첫째, 논문의 내용을 구조적으로 표현함으로써 논문 전체의 내용을 빠르고 정확하게 이해할 수 있다. 논문 전체를 다 읽어 볼 필요 없이 논문의 전체 구조의 이해와 알고자 하는 영역을 곧 바로 접근하여 파악 할 수 있다. 물론 자세한 내용을 원할 때는 논문의 원본을 참조하여야 한다. 둘째, 저자와 심사위원 간에 논문의 내용에 대하여 토론의 기회가 주어지기 때문에 공정한 토론을 이끌 수 있다. 이는 전통적인 저널 시스템에서 발생하는 일방적인 평가가 되는 것을 보완하여 공정한 심사가 가능하게 한다. 셋째, 심사활동에 편의를 제공한다. 논문에 대한 최종 평가에 있어서 심사위원과 저자 등의 토론 내용을 살펴보고 공정한 심사를 할 수 있으며, 편집장의 경우 각 심사위원의 심사 내용과 의견을 종합한 통계를 참고하여 최종 평가를 내리는데 도움을 준다. 넷째, DBMS 연계를 위해 JDBC를 사용함으로써 Oracle뿐만 아니라 JDBC Driver를 제공하는 모든 DBMS연계를 쉽게 할 수 있다. 다섯째, Object-Oriented 개념에 맞게 구현하였으므로 프로그램 source의 이해와 확장이 용이하다.

현재의 WebReview 시스템은 프로토타입 시스템으로 구현이 되었을 뿐 제2장에서 지정한 요구사항을 다 만족시키지 못하고 있어, 실용적인 시스템으로 발전시키시기까지는 다음과 같은

면에서 확장이 요구되고 있다. 첫째, 현재에는 논문 검증 단계를 제공하기 위해 attack과 defense만 지원되지만, 정보 요청, 전문가의 견해 등, 토론 시 필요한 정보가 충분히 공유될 수 있게 하기 위해서 멀티미디어 데이터를 지원하도록 하는 확장이 필요하다. 둘째, 논문의 요약을 위한 기본 구조가 issue-position-argument로 고정되어 있으나, 앞에서 지적한 바와 같이 연구분야의 특성에 맞는 다양한 표현구조에 대한 연구가 필요하다. 특별히 position 부분을 연구 방법, 연구결과 등 좀더 상세하게 표현을 할 수 있도록 composite structure로 전환하는 것을 고려해 볼 필요가 있다. 또한 저자와 심사위원들간의 대화가 현재는 argument에 대해서만 가능하지만, position을 상세하게 composite structure로 바뀌게 되면 position을 이루고 있는 상세 구조 하나 하나에 대해서도 대화를 통해서 공정한 심사가 이루어지도록 함이 바람직하다. 셋째, 심사위원의 선정과정을 도울 수 있도록 심사위원 pool을 데이터베이스화하는 게 필요하다. 넷째, 요약본의 작성 시 Web Editor와 같은 응용프로그램을 사용할 수 있도록 하여야 한다. 그리고, 논문게재 여부에 대한 편집장의 최종 결정을 도울 수 있는 기능에 대한 연구가 추가적으로 시스템에 반영되어야 한다. 현재 이상의 여러 가지 확장에 대하여 새롭게 구현하고 있다.

이러한 많은 확장의 여지가 있음에도 불구하고, 전통적인 저널 시스템의 몇 가지 단점을 보완할 수 있는 가능성에 대하여 시험 구현하였다는 점이 이 논문의 공헌이라 할 수 있다. 그리고 이상에서 언급한 확장이 앞으로 구현이 된다면 그 실용가치는 상당히리라 기대된다.

〈참고 문헌〉

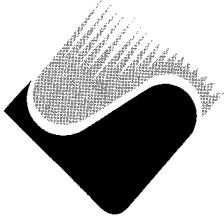
- [1] J. Conklin and M. Begeman, gIBIS: A Hypertext Tool for Exploratory Policy Discussion, *Computer Supported Cooperative Work 88*, Portland, OR, 1988.
- [2] H. Dehlinger and J-P Protzen, Debate and Argumentation in Planning: An Inquiry into Appropriate Rules and Procedures, *Working paper No. 178*, 1972.
- [3] Stephen P. Harter, "The Impact of Electronic Journals on Scholarly Communications: A Citation Analysis," *The Public-Access Computer Systems Review*, Vol. 7, No. 5, 1996.
- [4] S. R. Hiltz, *Online Communities: A Case of the Office of the Future*, Norwood, NJ: Ablex, 1984.
- [5] Brian R. Gaines, "An Agenda for Digital Journals: The Socio-Technical Infrastructure of Knowledge Dissemination," *Journal of Organizational Computing*, Vol. 7, No. 5, 1993, pp. 135-193.
- [6] R. McCall. PHIBIS: Procedurally Hierarchical Issue-Based Information Systems, *Proceedings of 1987 Conference on Planning and Design in Architecture*, American Society of Mechanical Engineers, 1987
- [7] Patrick Naughton, *THE JAVA HANDBOOK*, McGraw Hill, 1996
- [8] P. Oakeshott, "The 'Blend' experiment in electronic publishing," *Scholarly Publishing*, Vol. 17, No. 1, 1985, pp. 25-36.
- [9] C. Pott and G. Bruns, Recording the Reasons for Design Decisions, *MCC Technical Report STP-304-87*, 1987.
- [10] Gail L. Rein and Clarence A. Ellis, rIBIS: A Real-Time Group Hypertext System, *MCC Technical Report STP-095-90*, 1990.
- [11] H. Rittel, and M. Webber, "Dilemmas in a General Theory of Planning," *Policy Sciences*, Vol. 4, 1973.

〈참조 web sites〉

- [1] AMS, Bulletin of the American Mathematical Society, <http://www.ams.org/journals/bull>
- [2] CCT, Online Journal of Current Clinical Trials, <http://www.oclc.org/oclc/promo>
- [3] DTJ, Digital Technical Journal, <http://www.digital.com/info/DTJ/home.html>
- [4] PACS, The Public-Access Computer Systems Review, <http://info.lib.hu.edu/pacsrev.html>
- [5] PSY, Psycology, <http://cogsci.ecs.soton.ac.uk/~harnad/psyc.html>
- [6] Dan Suthers and Arlene Weiner, Groupware for Developing Critical Discussion Skills, <http://www-cscl95.indiana.edu/cscl95/suthers.html>

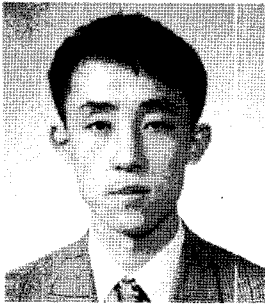
◆ 이 논문은 1998년 4월 30일 접수하여 1차 수정을 거쳐 1998년 8월 24일 게재 확정되었습니다.

◆ 저자소개 ◆



서 용 무 (Suh, Yongmoo)

서울대학교 사범대학 수학과를 졸업한 후, 한국과학원에서 전산학을 전공하였으며, KIST 전산센터 연구원을 지낸 후, University of Texas(Austin)에서 경영정보학으로 박사학위를 취득하였다. SDS, 세종대학교, 건국대학교를 거쳐 현재 고려대학교 경영대학에서 재직 중이다. 주요 관심분야로는 organizational computing, object-oriented database and paradigm 그리고, data mining 등이다.



전 건 호 (Jeon, Geonho)

고려대학교 전산과학과를 졸업하고 현대전자에서 근무하며 선박전적 시스템, 특허정보검색 프로젝트, 의료영상저장 및 전송 시스템(PACS) 개발 등을 수행하였다. 고대 경영대학원에서 경영학 석사(MBA) 학위를 취득하였으며, 현재 고려대학교 경영학박사(전공:MIS)학위 과정에 있다. 주요 관심분야는 객체지향기반 Computer Support for Group Process 등이다.