

## 기계부품의 효율적인 개발을 위한 정보서비스 및 공동작업환경의 지원

이명준

울산대학교 컴퓨터정보통신공학부

### <요 약>

최근 웹의 확산과 더불어 제품개발을 위해 공동작업 구성원들 사이의 정보 공유와 재사용을 지원하는 웹 기반의 정보시스템 개발에 대한 관심이 고조되고 있다. 공동작업 시스템에서는 의사교환 및 정보의 공유를 위해 설계문서, 기술문서 등의 전자 문서들이 많이 이용되고 있으므로 공동작업을 지원하기 위한 공유작업공간과 문서를 체계적으로 관리할 수 있는 통합적인 시스템을 개발하는 것이 바람직하다.

본 보고서에서는 웹 기반의 공동작업을 지원하기 위한 정보시스템의 설계와 구현에 대해 설명한다. 통합된 시스템은 *CoSpace*와 *CoDocs*라는 두 개의 독립된 시스템으로 구성되어 있다. *CoSpace*는 작업그룹 구성원들의 공동작업을 위해 공유작업공간을 지원하고, *CoDocs*는 개인작업 및 공동작업의 결과로서 생성되는 문서들을 체계적으로 관리할 수 있도록 하여준다. 통합된 시스템은 마이크로소프트 IIS 4.0 웹 서버를 사용하고, 시스템의 모든 상태를 저장하기 위해 SQL Server 6.5 데이터베이스를 사용한다. 시스템 사용자의 작업요청이 있을 때마다 발생하는 반복적인 데이터베이스 접속시도를 피하고 시스템의 성능향상을 위해 개발된 시스템은 일군의 데이터베이스 연결 설정을 유지하면서 적절히 작업요청에 따른 데이터베이스 연결을 할당하여 준다.

## Supporting Information Services and Collaborative Environment for Effective Development of Machine Parts

Myung-Joon Lee

School of Computer Engineering and Information Technology  
University of Ulsan

### <Abstract>

Recently, through the wide spread of WWW, interest in developing web-based information systems to support sharing or reusing information on product development among collaborating group members is highly increased. Since many electronic documents such as design documents and technical reports are used to exchange opinions and share information in *collaborative works*, it is desirable to build integrated systems which support *shared workspaces* for collaborative work as well as manage electronic documents systematically.

In this report, we discuss the design and implementation of a web-based information system for collaborative work, where two systems named *CoSpace* and *CoDocs* are effectively integrated. The CoSpace system supports shared workspaces for members of work groups, whereas the CoDocs system systematically manages electronic documents produced through personal or collaborative works. The integrated system uses MicroSoft IIS 4.0 web server, storing all of the system states in SQL Server 6.5 DBMS. To avoid frequent database connection/disconnection processes and enhance the performance of the developed system, the integrated system maintains a set of *database connections*, assigning those connections according to requests of the systemusers.

## 1. 서론

인터넷 사용에 대한 욕구가 기하급수적으로 증가하면서 웹(World Wide Web)에 관련된 신기술과 소프트웨어의 개발속도가 급격히 빨라졌다. 이러한 움직임은 통신망의 역할을 새롭게 변화 시키고 있으며, 또한 웹을 이용한 응용분야도 급속도로 확대되고 있는 추세이다.

웹은 이기종 시스템간의 접속문제를 해결할 뿐만 아니라 문서 표시 언어(Document Mark-up Language)의 지원, 저렴한 구축비용, 사용의 편리성, 확장의 용이성, 다양한 미디어를 지원할 수 있다는 장점을 가지고 있다[1]. 이러한 장점들은 기업 및 학교와 같은 조직내의 정보를 효율적으로 활용하기 위한 정보시스템(Information System)을 구축하는 새로운 방안(solution)이 되고 있다. 그리고 웹의 클라이언트 응용 프로그램인 브라우저(browser)는 GUI(Graphic User Interface)방식으로 인터넷 서비스를 통합적으로 제공하며, 거의 모든 컴퓨팅 환경에 포함되어 있다. 그러므로 사용자들은 자신이 속한 컴퓨팅 환경에 관계없이 웹 서비스가 가능한 곳이라면 어디에서라도 자신의 브라우저를 통하여 인터넷상의 모든 정보를 얻을 수 있다.

현재의 공동작업은 광범위하게 분산되어있는 경우가 많으며, 여러 작업그룹내의 구성원들은 이런 환경에서 실시간으로 서로의 정보를 교환하고 공유하기를 원한다. 그리고 작업그룹내의 구성원들은 서로 공유한 정보를 잘 관리하여 이후에도 재사용하기를 바란다. 또한, 현재의 공동작업시스템에서는 의사교환 및 정보의 공유를 위해 많은 문서들이 이용되고 있는데, 정보의 활용도를 높이고 체계적으로 문서들을 관리하기 위해서는 이러한 공동작업장 기능과 문서를 효율적으로 관리할 수 있는 통합적인 시스템이 요구된다. 본 연구에서는 이러한 기능을 제공하기 위한 시스템을 설계하고 구현하였다.

본 연구에서 구현한 시스템은 기계부품 설계에 있어 효율적인 정보공유를 제공하기 위해 공동작업을 지원하는 공동작업장(Shared Workspace) 시스템 'CoSpace'와 설계문서, 기술문서 등의 많은 기술문서를 체계적으로 관리하고 이를 바탕으로 다양한 정보서비스를 제공하는 문서관리 시스템 'CoDocs'로 구분되어 진다[17][18]. 공동작업장은 웹을 기반으로 이러한 공동작업을 지원하기 위한 기반 환경으로서 그룹의 구성원들이 접근가능하고 공유된 정보를 저장하는 저장소이다. 반면, 문서관리시스템은 개인작업 및 공동작업의 결과로서 생성된 기존의 다양한 기술문서들을 사용자가 체계화된 문서정보조직에 등록시키고 검색할 수 있도록 하는 정보서비스를 제공한다.

구성은 다음과 같다. 1장 서론에 이어 2장에서는 공동작업장 시스템 CoSpace에 대하여 설명하고, 3장에서는 문서관리시스템 CoDocs에 대해서 설명한다. 그리고 4장에서는 두 시스템의 통합에 대하여 기술하고 5장에서는 관련연구에 대하여 알아본다. 마지막으로 6장에서는 결론 및 향후 연구방향에 대하여 언급한다.

## 2. CoSpace 공동작업 시스템

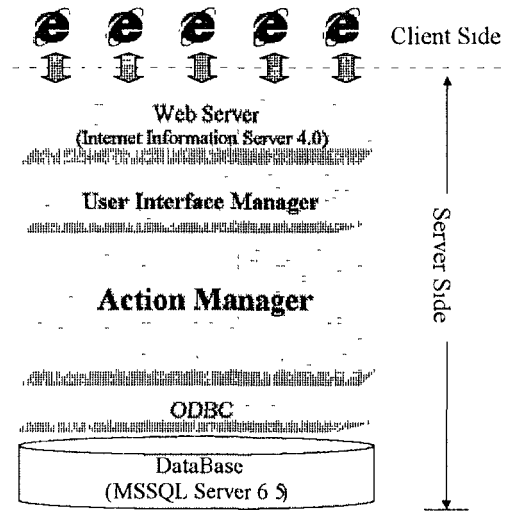
기계 부품의 개발에 있어서 여러 사람이 효과적으로 공동작업을 하게 된다면 개발의 효율성과 생산성을 향상시킬 수 있을 것이다. 하지만 공동작업을 위한 기술문서와 정보를 공동작업의 참여자들이 효과적으로 공유하는 것은 쉽지 않다. 공동작업 환경에서 사용자들이 효율적으로 작업을 진행하기 위해서는 다양한 기술문서를 효과적으로 공유하는 것이 바람직하다. 본 연구에서는 효과적인 공동작업을 지원하기 위해 CoSpace 공동작업 시스템을 구현했다. CoSpace 공동작업 시스템은 모든 사용자들이 쉽게 사용할 수 있도록 현재 널리 이용되고 있는 표준 웹 브라우저를 통하여 정보를 공유하고 관리할 수 있도록 설계하고 구현했다.

### 2.1 CoSpace 공동작업 시스템의 설계

본 절에서는 CoSpace 공동작업 시스템의 구조와 CoSpace를 구성하고 있는 사용자 인터페이스 관리자, Action 관리자에 대해 설명하도록 하겠다.

#### 2.1.1 CoSpace 공동작업 시스템의 구조

CoSpace 공동작업 시스템은 모든 작업이 서버에서 수행되는 중앙집중제어방식의 응용 프로그램으로서, 대부분의 작업이 데이터베이스와 연동되어 수행이 되는 방식으로 구현하였다. 먼저 CoSpace 공동작업 시스템의 구조는 크게 클라이언트와 서버로 나누어 질 수 있다. 클라이언트는 별도의 부가적인 응용프로그램이 없이 단지 웹 브라우저로 CoSpace 공동작업 시스템의 서버로 접속할 수 있다. 서버 쪽의 구조는 아래 그림과 같이 크게 5가지로 나누어진다. 웹서버, 사용자 인터페이스 관리자(User Interface Manager), Action 관리자(Action Manager), ODBC(Open Database Connection), 데이터베이스(Database)로 구성되어진다.



[그림 2-1. CoSpace 공동작업 시스템구조]

웹서버는 WWW를 통해서 접근할 수 있는 모든 자료들을 보관, 관리, 통제하는 프로그램이다. 웹 브라우저를 통해 사용자가 어떠한 서비스를 요청했을 때 제일 먼저 웹서버에 접근하게 된다.

사용자 인터페이스 관리자는 사용자의 작업요청을 분석하여 어떤 Action과 이벤트(Event)가 발생했는지 조사한 후 TCP/IP 소켓(socket)를 통하여 Action 관리자에게 작업 요청을 보낸다. Action 관리자가 요청한 서비스에 대해 작업을 수행한 후 그것에 대한 결과값을 다시 TCP/IP 소켓(socket)를 통하여 사용자 인터페이스 관리자에게 보내게 된다. 그러면 사용자 인터페이스 관리자는 그 결과값을 토대로 서비스를 요청한 사용자의 화면에 출력되어질 HTML 문서, 즉 사용자 인터페이스를 만들어서 사용자에게 웹서버를 통해 사용자에게 보내지게 된다.

Action 관리자(Action Manager)는 클라이언트에서 작업요청이 있을 때마다 데이터베이스와 연결을 시도해야 되는 부담 때문에 서버의 부하가 커지는 단점이 있는데 이를 해결하기 위하여 항상 데이터베이스와 연결포인터(Connection Pointer)를 유지하고 있다. 클라이언트의 모든 사용자 서비스 요청에 대한 실제적인 작업은 Action 관리자가 데이터베이스와 연동하여 작업을 수행하게 된다.

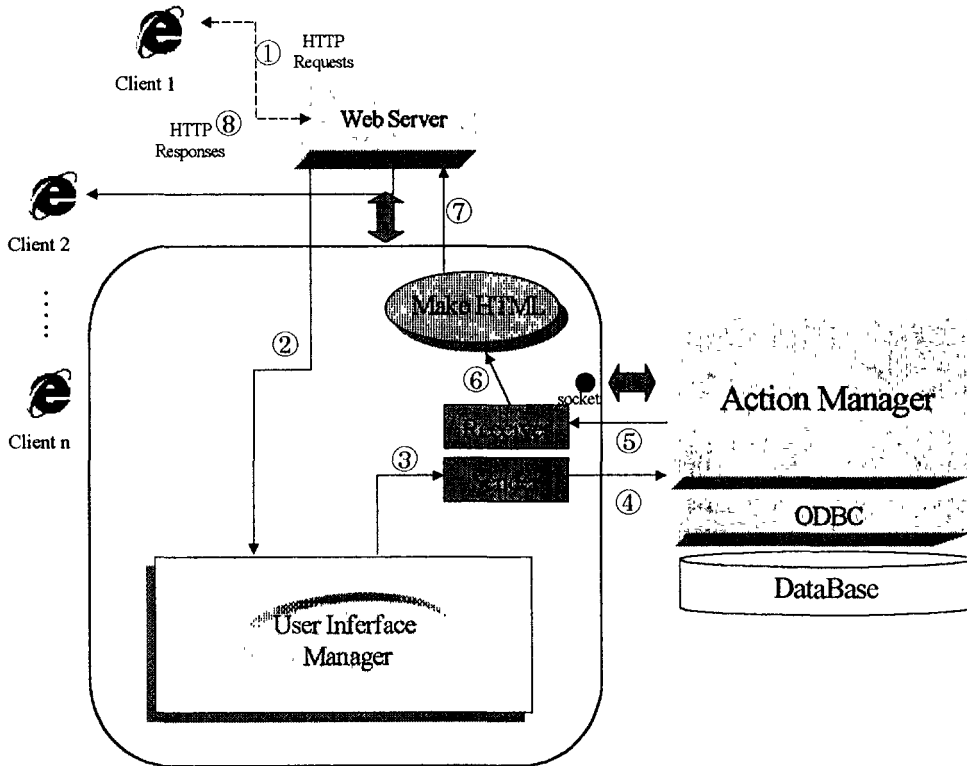
## 2.1.2 CoSpace 공동작업 공간의 구성요소

CoSpace 시스템은 구조적으로 크게 사용자의 웹 브라우저와의 연동을 통하여 작업을 수행하는 User Interface Managers 와 실질적으로 작업을 처리해서 그 결과를 다시 사용자에게로 돌려주는 Action Manager로 구성되어 있다. 그리고 이러한 시스템의 모든 시스템 상태 정보는 데이터베이스에 저장되어 있고 이 정보는 Action Manager의 작업에 의해 추가되거나 수정될 수가 있다.

### 2.1.2.1 사용자 인터페이스 관리자(User Interface Manager)

User Interface Manager는 웹서버와 Action Manager 사이에서 존재하며 사용자들의 웹 브라우저 내에서 CoSpace 시스템에 관련된 모든 사용자 인터페이스를 보여준다. 또한

MicroSoft IIS 4.0 웹서버를 통해 사용자로부터 Action을 받아들이고 이렇게 받아들인 사용자 Action 정보를 Action Manager로 전달해주는 역할을 하게 된다. Action Manager가 사용자로부터 받아들인 Action에 대해 해당하는 작업을 처리해주고 그 결과를 사용자 인터페이스 관리자로 넘겨주게 되고 사용자 인터페이스 관리자는 그 결과를 사용자의 웹 브라우저로 웹 서버를 통해 새로운 웹 페이지 형태로 전달해주게 된다. 이러한 과정을 그림으로 보면 다음과 같다.



[그림 2-2. 사용자 인터페이스 관리자의 구조]

위의 과정을 좀 더 자세히 살펴보면, 사용자는 자신의 웹브라우저에서 원하는 작업을 위하여 작업 창을 띄우고 해당하는 작업을 선택하여 사용자 인터페이스 관리자에게 Action 요청을 하게 된다. (그림 2-3 참조)



[그림 2-3. Action Window]

사용자 인터페이스 관리자는 사용자가 웹 브라우저로부터 만들어낸 Action에 대해 작업 요청을 Action Manager에게 보내기 위해 미리 정해진 규칙대로 메시지를 만들어 소켓을 통하여 보내게 된다. 이때 정의되는 메시지 정보의 형태는 다음과 같다.

Length	작업형태	사용자아이디	작업데이터
--------	------	--------	-------

이렇게 보내진 작업 요청은 Action Manager에 의해 처리되고 Action Manager는 그 결과에 대한 정보와 작업 수행 성공 여부에 따라 각기 다른 형태의 작업 결과를 HTML 형태의 스트림으로 사용자 인터페이스 관리자에게 보내주게 된다. 사용자 인터페이스 관리자는 Action Manager로부터 온 작업 결과를 분석하여 해당 페이지에 대한 정보만을 웹 서버에게 보내주어 사용자에게 전달하게 된다.

사용자 인터페이스 관리자에 대한 추상적인 알고리즘 기술은 다음과 같다.

```

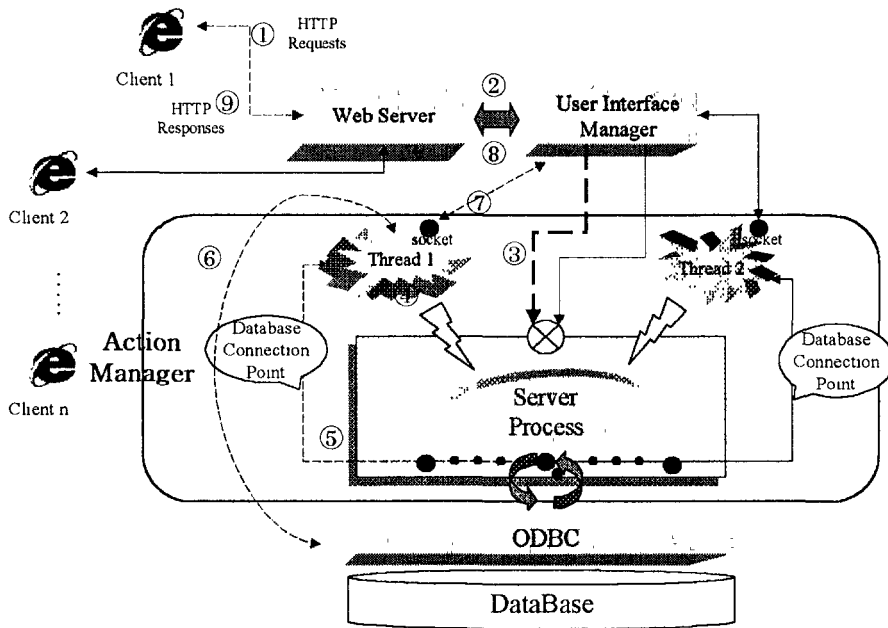
main {
    create the socket;
    binding socket;
    create packet;
    send to ServerSocket;
    receive packet;
    tokenize packet and make user interface;
}

```

### 2.1.2.2 Action Manager

#### 가. Action Manager 구조

클라이언트에서 작업요청이 발생하면, 사용자 인터페이스 관리자(User Interface Manager)는 TCP/IP 소켓(socket)을 통해 Action Manager의 서버프로세스에게 접속한 후 자신의 정보를 전달하고, Action Manager는 사용자 서비스 요청이 소켓(socket)을 통해 들어올 때 마다 쓰레드(thread)를 만들어 자신이 연결 유지하고 있는 데이터베이스의 연결포인트와 현재 연결되어진 소켓정보를 쓰레드에게 전송한다. 쓰레드는 이 두 정보를 기반으로 소켓으로부터 정보를 읽는다. 어떤 Action을 요청했는지를 들어온 정보를 분석한다. 이 정보들을 기반으로 데이터베이스와 연동하여 작업을 수행하고 그 결과를 쓰레드는 TCP/IP 소켓을 통해서 사용자 인터페이스 관리자(User Interface Manager)에게 전달한다. 아래의 그림은 Action 관리자의 내부 구조를 나타낸 것이다.



[그림 2-4. Action Manager 구조]

#### 나. 작업객체(Object)의 정의 및 종류

CoSpace의 사용자들이 개인작업공간 또는 그룹작업공간에서 관리하고 공유할 객체들을 총칭해서 작업객체(Object)라고 정의하였다.

- (1) 폴더 : 작업객체 관리를 위한 목적으로 여러 개의 작업객체들을 저장할 수 있는 디렉토리
- (2) 문서 : 여러 가지 형태의 문서, 기술문서, 일반문서 등등
- (3) URL : 웹상의 자원의 위치
- (4) 공고 : 간단한 글 또는 그룹의 구성원들에게 보여줄 공지사항, 공지 내용의 중요도에 따라서 등급을 주어 게시할 수 있다.

#### 다. Action 정의 및 종류

- (1) 폴더추가 : 여러 개의 작업객체들을 저장할 수 있는 디렉토리를 만드는 기능이다.
- (2) 문서추가 : 여러 종류의 파일들을 작업공간에 저장하는 기능이다.
- (3) URL추가 : 웹 상에서 참고하고자 URL를 공유작업공간에 추가하는 기능이다.
- (4) 공고추가 : 공지사항이나 간단한 메모를 공유작업공간에 추가하는 기능이다.
- (5) 삭제 : 객체들을 삭제하는 기능이다.
- (6) 이름바꾸기 : 객체의 이름을 변경하는 기능이다.
- (7) 잘라내기 : 작업공간에서 작업개체를 잘라내어 임시영역에 저장하는 기능이다.
- (8) 복사하기 : 작업공간에서 작업개체를 복사하여 임시영역에 저장하는 기능이다.
- (9) 붙여넣기 : 임시영역에 저장된 데이터를 실제 작업공간에 저장하는 기능이다.
- (10) 그룹에추가 : 폴더추가, 문서추가, URL추가, 공고추가와 같은 작업을 통해 자신의 작업공간에 등록되어진 폴더, 문서, URL, 공고 등을 사용자가 참가하고 있는 그룹에 자신의 작업객체들을 추가하여 다른 사용자들과 공유하고자 할 때 사용된다.
- (11) 그룹생성 : 개인작업공간에서 공동작업 또는 정보공유를 목적으로 하는 가상의 그룹작업공간을 생성하는 기능이다.
- (12) 그룹참가 : 개인작업공간에서 사용자가 그룹에 능동적으로 참가하기 위한 기능이다.
- (13) 그룹로깅 : 현재참가중인 그룹에 로깅하는 기능이다. 로깅하기를 원하는 그룹을 마우스로 선택함으로써 이 기능을 수행할 수 있다.
- (14) 그룹탈퇴 : 현재참가중인 그룹에서 탈퇴하는 기능이다. 현재 참가중인 그룹에서 탈퇴하기를 원하는 그룹을 마우스로 선택함으로써 이 기능을 수행할 수 있다.
- (15) 그룹삭제 : 그룹이 불필요할 때 그룹을 삭제하는 기능이다. 그리고 그룹의 목적에 어긋나게 운영되는 그룹은 CoSpace의 총괄 책임자가 강제로 퇴출시킬 수 있는 기능이 있다.

#### 라. Event 정의 및 종류

CoSpace 공동작업 시스템에서의 이벤트는 여러 가지가 있다. 먼저 폴더추가, 파일추가, URL추가, 공고추가, 이런 Action들을 하게되면 반드시 만든 날짜와 시간이 출력되게 되어 있다. 또 그룹작업공간인 경우에는 그룹의 멤버 중에 누가 언제 추가하였는지 이벤트가 화면에 나타나게 되어있다. 여러 가지 이벤트는 단순화하여 다음과 같이 세 가지로 구분하였다.

- (1) New : 그룹작업공간에서 작업개체들이 생성되었을 때 보여주는 이벤트.
- (2) Change : 그룹작업공간에서 기존에 존재하고 있던 작업개체에 변화가 일어났을 경우 보여주는 이벤트이다.
- (3) Read : 그룹작업공간에서 작업개체를 누군가에 의해 읽혀지거나, 다운로드(download) 되었을 때 보여주는 이벤트이다.

## 2.2 CoSpace 시스템의 구현

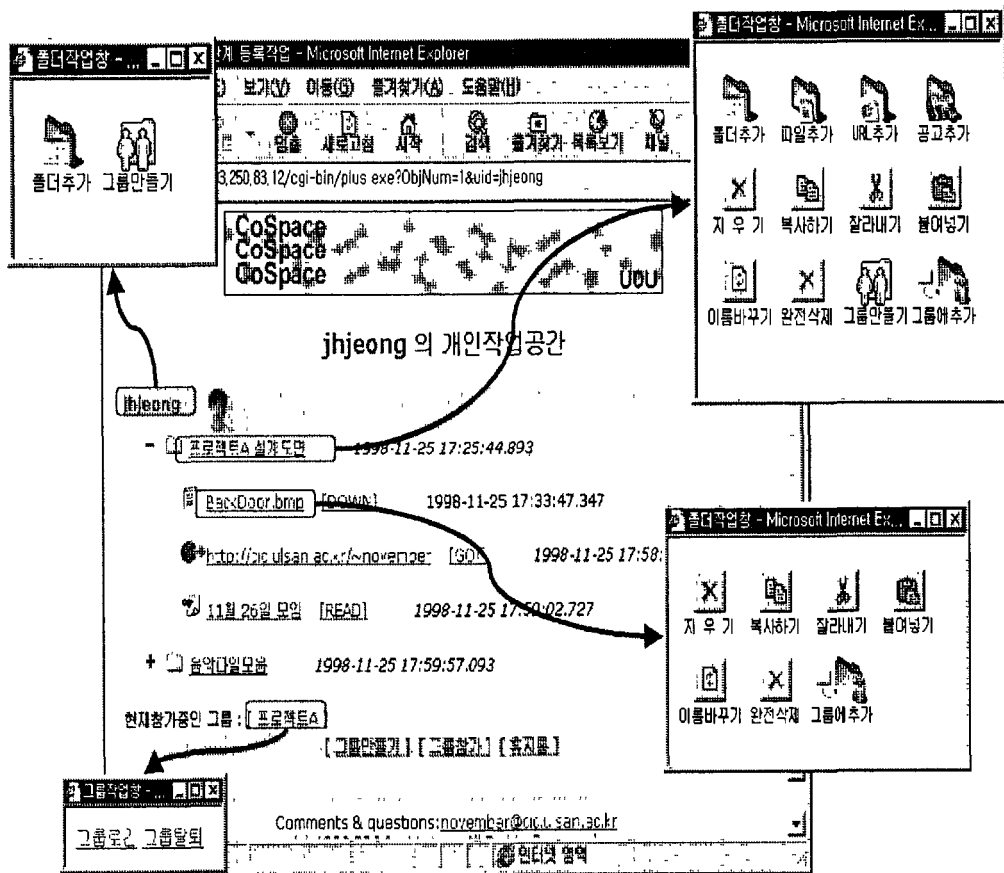
CoSpace의 작업공간은 개인 또는 그룹을 위하여 기술문서의 관리와 공유를 위한 가상 공간으로서 윈도우즈의 탐색기와 유사한 형태로 제공되며 개인작업공간과 공동작업공간으로 나누어진다.



### 2.2.1 개인작업공간

CoSpace 공동작업 시스템에 등록을 한 모든 사용자에게 제공되는 작업공간으로 자신의 작업객체(Object)를 관리할 수 있는 가상 작업공간이다. 가상 작업공간에는 다양한 형태의 정보(기술문서, 일반문서, 특정URL, Article)를 등록하고 관리할 수 있게 구현했다. 등록된 정보는 폴더 계층구조로 사용자에게 보여지도록 되어 있다. 구성원들은 개인작업공간에 자신의 지역(local)에 있는 정보를 작업공간에 저장시킬 수 있으며, 작업공간에 있는 정보를 자신의 지역시스템(local system)으로 다운로드 할 수도 있다.

정보를 공유하기 위해서 모든 사용자들은 작업의 특성에 따라 그룹을 생성할 수 있으며, 이미 만들어져 있는 그룹에 참가 할 수도 있다. 아래의 그림은 실제 개인작업장의 화면 구성을 나타낸 것이다.



[그림 2-5 개인작업공간]

그룹을 만든 사용자는 자동적으로 그 그룹의 마스터가 되고 나머지 그룹의 참가나 초청으로 들어오게 되는 사용자들은 멤버가 된다. 그룹의 멤버를 구성하는 방법은 두 가지 형태로 지원해 준다. 첫 번째, 그룹의 특성에 따라 그룹내의 마스터나 모든 구성원 멤버들이 그룹에 속해 있지 않은 다른 사용자들에게 그룹의 정보를 공유할 수 있도록 초청을 하는

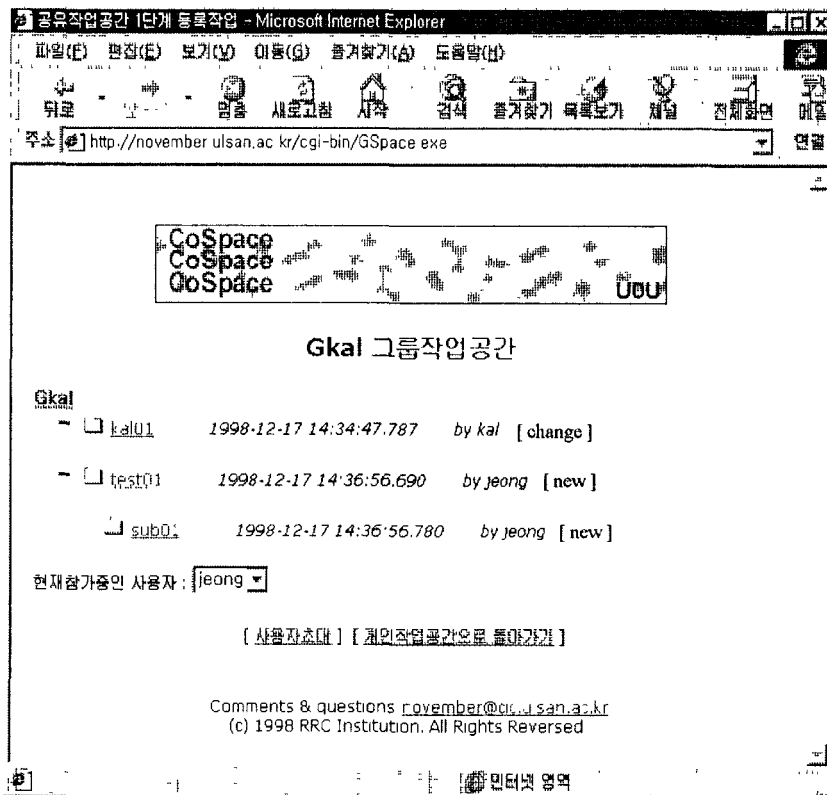
수동적인 방법이다. 초청을 받은 사용자는 CoSpace 공동작업 시스템에 접속할 때 어떤 그룹으로부터 초청을 받았는지 메시지가 출력되고 사용자는 그 초청에 대해서 승인할 수도 있고 거절할 수도 있는 권한이 있다. 승인 시에는 그 순간부터 그 그룹의 멤버가 되고 언제든지 그 특정한 그룹에 들어갈 수 있다. 두 번째, 어떤 특정한 그룹의 멤버가 아닌 사용자가 능동적으로 자신이 원하는 그룹에 참가하고자 할 경우, 사용자는 그 그룹의 참가신청을 하고 그 그룹의 특성에 따라 마스터나 멤버가 승인 또는 거절함으로써 그 그룹에 참가할 수 있게 된다.

본 연구에서는 그룹의 특성에 따라 아래와 같이 세 가지 형태의 그룹을 지원한다.

1. 그룹의 책임자만 다른 사용자를 그룹의 멤버로 초청할 수 있는 그룹
2. 그룹의 멤버면 누구나 다른 사용자를 그룹의 멤버로 초청할 수 있는 그룹
3. 그룹의 모든 멤버가 동의할 경우에만 다른 사용자를 그룹의 멤버로 초청할 수 있는 그룹

### 2.2.2 공동작업공간

공동작업 또는 그 외의 용도로 정보의 공유를 목적으로 하는 가상의 작업공간이다. CoSpace 공동작업 시스템에 개인작업공간을 가지고 있는 사용자들이면 누구나 만들 수 있고, 작업의 특성에 따라 그룹 종류 중에 적합한 그룹을 선택, 관리 할 수 있다. 이 그룹작업공간에서 개인의 정보를 공유하게 된다. 아래의 그림은 실제 그룹공동작업장의 화면 구성을 나타낸 것이다.



[그림 2-6. 그룹작업공간]

### 2.2.3 구현환경

본 연구에서는 운영체제로 Window NT 서버와 마이크로소프트사의 IIS (Internet Information Server) 4.0 웹 서버를 사용하였고, 데이터베이스로는 MS SQL Server 6.5를 사용했다. 또한 Action Manager는 ODBC 인터페이스를 사용했으므로 ODBC를 지원하는 모든 데이터베이스와 연결이 가능하다[7][8]. 사용자 인터페이스 관리자(User Interface Manager)는 C언어로 구현했고, 서버프로세스는 C++를 사용하여 구현하였다.

## 3. CoDocs 문서관리 시스템

CoDocs 문서관리 시스템은 개인작업 및 공동작업의 결과로서 생성되는 문서들을 사용자가 문서정보조직에 등록시키고 검색할 수 있도록 함으로써 효율적으로 문서들을 관리하기 위한 시스템이다.

### 3.1 CoDocs 문서관리시스템의 설계

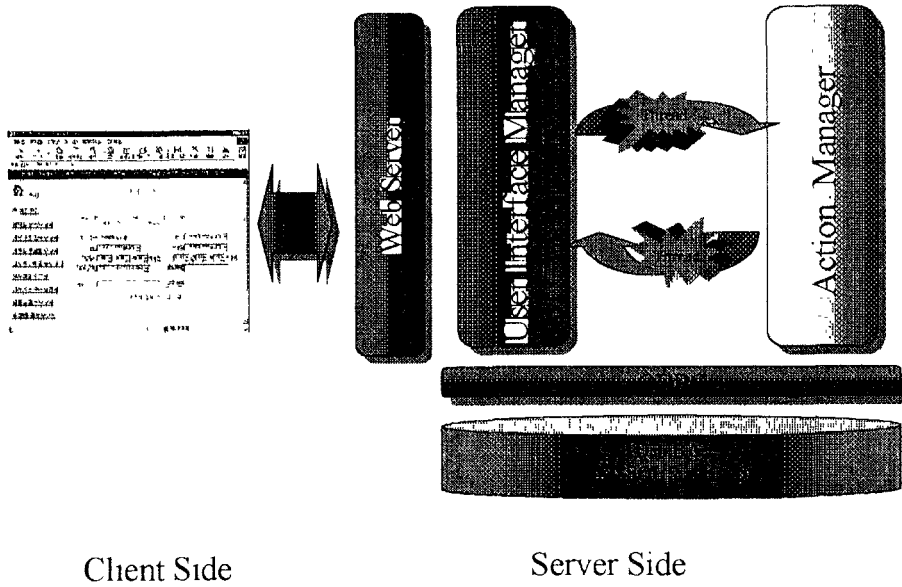
본 절에서는 문서관리시스템의 전체적인 구조와 시스템의 구성요소, 그리고 이를 구현하기 위해 사용된 데이터베이스의 설계에 대해서 알아본다.

#### 3.1.1 CoDocs 문서관리시스템의 구조

CoDocs 문서관리시스템은 모든 작업이 서버에서 수행되는 중앙집중제어방식의 응용프로그램으로서, 대부분의 작업이 데이터베이스와 연동되어 수행이 되는 방식으로 구현하였다. 먼저 CoDocs 문서관리시스템의 구조는 크게 클라이언트부분과 서버부분으로 나누어진다. 클라이언트는 별도의 부가적인 응용프로그램이 없이 단지 웹 브라우저로 CoDocs 문서관리시스템의 서버부분으로 접속할 수 있다. 서버 쪽의 구조는 아래 [그림 3-1]과 같이 크게 5가지로 나누어진다.

먼저 웹서버는 클라이언트측의 작업요청정보를 서버측 응용프로그램에 전달하고 처리결과를 웹브라우저로 전송하는 역할을 한다. 사용자 인터페이스 관리자(User Interface Manager)는 사용자의 작업요청을 분석하여 어떤 작업이 요청되었는지 조사한 후 TCP/IP 소켓(socket)을 통하여 Action Manager로 전달하고 처리결과를 토대로 서비스를 요청한 사용자의 화면에 출력되어질 HTML 문서, 즉 사용자 인터페이스를 만들어서 사용자에게 웹서버를 통해 사용자에게 보내는 역할을 한다.

시스템에서 발생하는 대부분의 작업은 데이터베이스 관련작업이므로 서버쪽 응용 프로그램은 클라이언트에서 작업요청이 있을 때마다 데이터베이스와 연결을 시도해야 되는 부담 때문에 서버의 부하가 커지는 단점이 있다. 이를 해결하기 위하여 Action Manager에서는 항상 데이터베이스와 연결포인터(Connection Pointer)를 유지하고 있다. Action Manager는 사용자 인터페이스 관리자로부터의 작업 요청을 처리하기 위해, ODBC API를 사용하여 데이터베이스작업을 처리하고 그 결과를 사용자 인터페이스 관리자로 넘겨주는 역할을 한다.



[그림 3-1] CoDocs 전체 시스템 구조

### 3.1.2 CoDocs 문서관리시스템의 구성요소

위에서 잠시 살펴본 것처럼 CoDocs 문서관리시스템의 구성요소는 크게 요청된 작업결과를 사용자에게 보여줄 인터페이스를 만들어 주는 User Interface Manager와 사용자로부터의 작업요청에 대한 데이터베이스 작업을 처리하는 Action Manager로 구성된다.

#### 3.1.2.1 사용자 인터페이스 관리자(User Interface Manager)

사용자 인터페이스 관리자는 사용자들로부터 작업요청을 받아 들여 시스템의 Action Manager에게 작업수행을 요청하고 요청한 작업에 대한 결과를 Action Manager로부터 넘겨 받아 정리하여 사용자에게 보여줄 인터페이스를 만드는 역할을 한다. 사용자 인터페이스 관리자는 아래와 같은 일련의 작업순서를 가지며 수행된다.

- (1) 웹서버로부터의 파라미터분석
- (2) Action Manager로 작업요청을 위한 메시지 생성
- (3) Action Manager와 통신을 위한 소켓생성
- (4) Action Manager로 작업요청 메시지 전송
- (5) Action Manager로부터 작업결과 메시지 수신
- (6) 작업결과를 바탕으로 사용자 인터페이스 생성

사용자 인터페이스는 Action Manager와 통신을 위해 TCP/IP 소켓을 생성하고 작업요청을 위하여 메시지를 생성하는데 Action Manager로 전송하는 메시지의 구조는 아래와 같다.

ActionName	UserId	TabName	ActionType	ActionComment1	...	ActionComment5
------------	--------	---------	------------	----------------	-----	----------------

[ 작업요청 메시지 ]

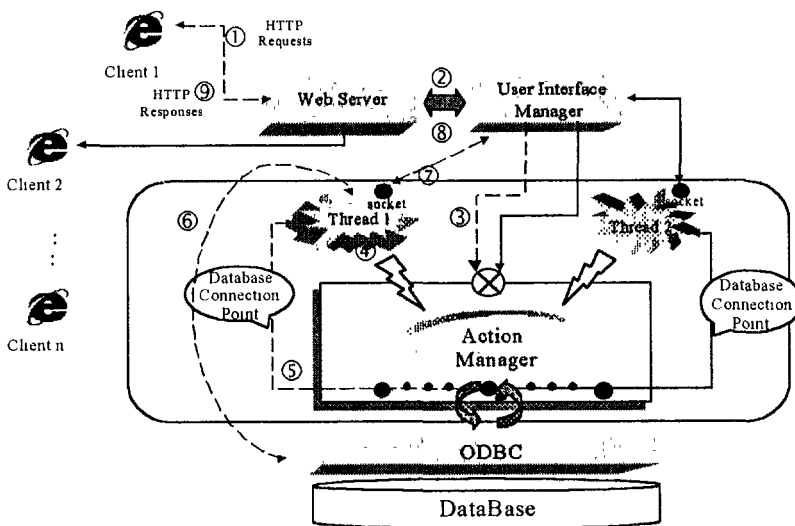
- ActionName: 작업요청에 대한 작업이름이 명시된다. ex) Upload, Search, ...
- UserId: 현재 작업중인 사용자의 아이디로서 인증을 위해 사용된다.
- TabName: 작업에 관련된 데이터베이스 테이블이름을 명시한다.
- ActionType: 요청된 작업의 세부작업 종류를 명시한다. ex) update, delete, ...
- ActionComment1 ... 5: 작업시에 필요한 기타정보를 포함한다.

### 3.1.2.2 Action Manager

본 절에서는 Action Manager의 전체적인 구조와 동작설명, 그리고 각 작업별로 간단한 기능을 설명한다.

#### 3.1.2.2.1 Action Manager 구조

클라이언트에서 작업요청이 발생하면, 사용자 인터페이스 관리자(User Interface Manager)는 TCP/IP 소켓(socket)을 통해 Action Manager의 서버프로세스에게 접속한 후 자신의 정보를 전달하고, Action Manager는 사용자 서비스 요청이 소켓(socket)을 통해 들어올 때 마다 쓰레드(thread)를 만들어 자신이 연결 유지하고 있는 데이터베이스의 연결포인터와 현재 연결되어진 소켓정보를 쓰레드에게 전송한다. 쓰레드는 이 두 정보를 기반으로 소켓으로부터 정보를 읽는다. 어떤 작업을 요청했는지를 들어온 정보를 분석한다. 이 정보들을 기반으로 데이터베이스와 연동하여 작업을 수행하고 쓰레드는 그 결과를 TCP/IP 소켓을 통해서 사용자 인터페이스 관리자(User Interface Manager)에게 전달한다.



[그림 3-2 Action Manager 구조]

### 3.1.2.2.2 Action Manager의 동작

Action Manager는 사용자 인터페이스 관리자로부터 작업요청 메시지를 받아 들어 요청된 작업을 처리하고 결과 메시지를 사용자 인터페이스 관리자로 소켓을 통하여 전달하는 역할을 한다. Action Manager는 아래와 같은 일련의 작업순서를 가지며 수행된다.

- (1) 기본 데이터베이스 테이블 생성 및 데이터베이스 연결 포인터 유지
- (2) 사용자 인터페이스 관리자로부터 작업요청시 새로운 소켓 및 쓰레드 생성
- (3) 작업요청 메시지분석
- (4) 작업을 요청한 사용자에게 대한 인증
- (5) 요청된 작업 처리 및 결과메시지 생성
- (6) 메시지전송 및 쓰레드 삭제

### 3.1.2.2.3 작업의 종류

Action Manager에서는 작업요청 메시지를 통하여 전달된 파라미터를 분석하여 작업의 종류를 구분하고 이에 해당하는 작업을 수행하도록 되어있다. 따라서 Action Manager에서는 작업의 종류에 따라 모듈별로 작업수행루틴을 정의하고 있다. 다음에서는 Action Manager에서 처리하는 작업의 종류와 기능에 대하여 살펴본다.

#### (1) 문서분류 추가/변경

문서분류 추가/변경 기능은 시스템의 문서 정보조직을 체계적으로 관리하기 위하여 계층적으로 문서분류를 구성한다.

#### (2) 문서등록

문서등록 기능은 사용자가 작업한 파일을 문서정보와 더불어 서버로 등록하는 기능이다. 이때 문서정보는 문서의 보안레벨에 대한 정보를 포함한다.

#### (3) 문서검색

문서검색 기능은 사용자가 문서정보를 이용하여 이미 등록된 문서를 검색하는 기능이다. 빠르고 정확한 검색을 위하여 다양한 검색조건이 제공되는데 이 검색조건들은 문서등록시에 저장된 문서 정보를 기반으로 한다 그리고 문서는 모든 사용자에게 보여지는 것은 아니며 접근 권한이 허용된 사용자에게만 선별적으로 보여 진다.

#### (4) 분류별 문서검색

분류별 문서검색 기능은 문서분류를 통하여 자신이 찾고자하는 문서를 찾을 수 있는 기능을 제공한다. 시스템 관리자는 이 기능을 통하여 등록된 문서를 삭제하거나 문서의 정보를 변경할 수 있다.

#### (5) 자료실

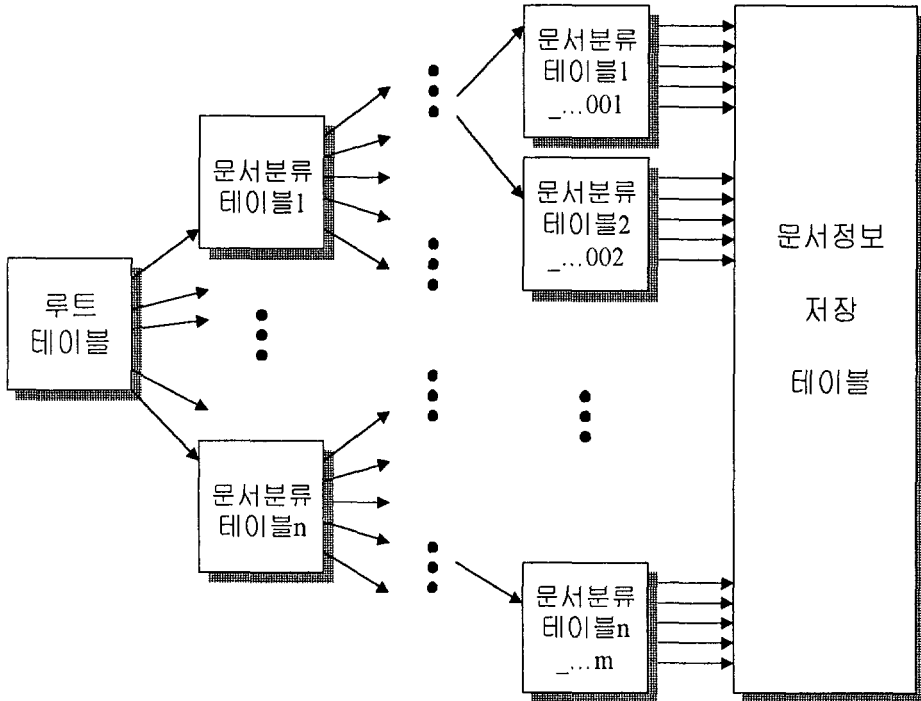
자료실 기능은 문서를 자신의 사용자 그룹이외의 다른 사용자들과 문서를 공유하기 위하여 게시하는 기능이다.

#### (6) 관리정보등록

관리정보 등록 기능은 문서종류 저장기능, 시스템의 사용자정보를 관리하기 위한 기능, 그리고 계층적인 사용자그룹을 관리하기 위한 사용자그룹 관리기능으로 구성되어 있다.

### 3.1.3. CoDocs 문서관리시스템의 데이터베이스 설계

본 연구에서는 사용자가 문서정보조직에 등록된 다양한 기술문서를 쉽게 이용하고 관리자가 문서의 문서정보조직을 체계적으로 관리하기 위한 데이터베이스를 설계하였다. 아래 [그림 3-3]은 계층화된 정보조직을 구성하기 위한 테이블 구조를 보여준다. 문서정보조직의 최상위 계층에는 루트 테이블이 존재하며 이 테이블에는 문서정보조직의 주 분류에 해



[그림 3-3] 문서분류를 표현하기 위한 데이터베이스 테이블 구조

당하는 내용이 저장된다. 루트 테이블의 하위계층에는 문서분류라고 하는 중간노드 테이블이 존재하는데 이 테이블들은 정보조직의 필요에 따라 서버쪽 응용 프로그램에 의해 동적으로 생성 및 삭제되면서 계층적인 테이블 구조를 형성한다. 이렇게 생성된 테이블 구조에서 문서정보는 최하위 계층의 문서분류에 링크되고 사용자는 이를 바탕으로 문서검색 및 관리 작업을 쉽게 수행할 수 있다. [그림 3-3]의 문서정보조직을 구성하는 각 노드테이블의 내용은 다음과 같다.

- Item\_id: 문서분류의 아이디.
- Item: 문서분류의 이름.
- Description: 문서분류의 설명
- ReadMe: 문서분류 도움 문서경로.
- Ptable: 문서분류의 부모테이블 이름.
- Ctable: 문서분류의 자식테이블 이름.

문서정보조직 외의 시스템의 기본적인 테이블로는 조직의 계층적인 구조를 표현하기 위한 사용자그룹 테이블과 사용자 인증을 위한 사용자정보 테이블, 부서간 자료 공유를 위한 자료실정보 테이블, 그리고 게시판 테이블 등으로 구성되어 있다.

### 3.2 CoDocs 문서관리시스템의 구현

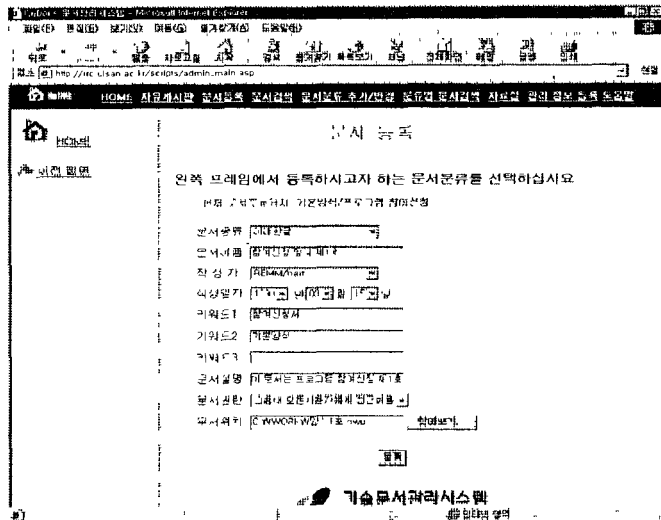
본절에서는 구현된 CoDocs 문서관리 시스템의 각 기능과 사용자 인터페이스에 대하여 설명한다.

#### 3.2.1 문서분류 추가/변경

문서관리시스템은 문서를 체계적으로 관리함으로서 정보의 활용성과 업무의 효율을 높이기 위해 구축된 시스템이다. 문서를 체계적으로 관리하기 위하여 본 연구에서 구현된 시스템은 문서정보조직을 계층적인 구조로 구성하고 있다. 문서정보조직은 문서분류라고 하는 중간노드를 이용하여 계층적으로 구성되는데, 문서분류는 필요에 따라 동적으로 생성 및 삭제되도록 하였다. 이렇게 구축된 문서정보조직을 바탕으로 사용자는 문서분류에 적합한 문서를 문서 정보와 더불어 서버에 등록하며, 이 정보를 이용하여 사용자는 찾고자하는 문서를 빠르고 정확하게 검색할 수 있다.

#### 3.2.2. 문서 등록

문서등록은 자신이 작성한 문서를 문서 정보와 더불어 서버에 등록하여 권한을 가진 사용자가 열람할 수 있도록 하는 기능을 말한다[2]. 문서정보는 데이터베이스에 저장되며, 이때 저장되어지는 문서정보의 항목들은 시스템 설치 시에 정의된다. 등록된 문서의 정보는 검색작업 시에 사용자가 원하는 문서를 쉽게 찾을 수 있게 하는 정보로서 이용된다. 또한 등록되는 문서는 문서권한에 대한 추가정보를 가지는데 이 정보는 사용권한을 갖지 않은 사용자로부터 문서를 보호하기 위한 보안기능의 자료로서 이용된다. 이때 추가되는 문서권한은 총 5등급으로 구분되며, 사용자의 문서요청에 대하여 접속한 사용자가 속한 사용자 그룹과 해당 문서의 권한 등급이 일치하면 접근이 허용된다.

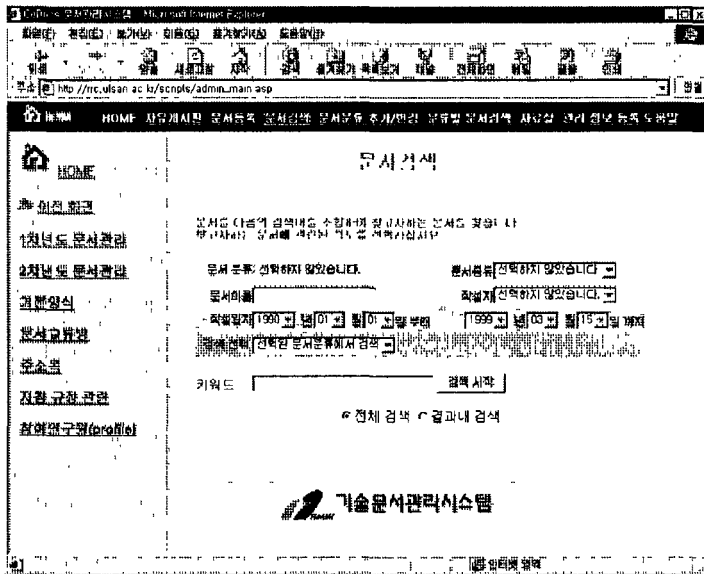


[그림 3-4 문서등록 화면]



### 3.2.3. 문서 검색

문서등록을 이용하여 문서정보조직에 저장되는 문서는 문서의 특징에 맞게 분류되어 저장된다. 이렇게 구축되어진 문서정보조직은 문서검색을 통하여 사용자가 원하는 문서를 정확하고 빠르게 찾을 수 있도록 지원한다. 문서검색은 일반적인 검색어를 이용하는 것보다 문서등록 시에 저장된 문서관리 정보를 검색 메뉴를 돕음으로서 검색을 보다 빠르고 편리하게 할 수 있도록 하였다. 검색된 문서에 대해서는 사용자가 다운로드 받아 볼 수 있는데, 모든 문서가 공개용은 아니므로 문서검색 결과로서 나타나는 문서는 권한이 있는 사용자만이 접근할 수 있도록 하였다. 이러한 기능은 문서 등록 시에 저장된 문서권한 정보에 의하여 지원된다.



[그림 3-5 문서검색 화면]

### 3.2.4 관리정보 등록

관리정보 등록에서는 문서등록 및 문서검색 작업시에 현재 시스템에서 사용하고 있는 문서의 종류를 등록할 수 있는 기능과 시스템에 등록된 사용자 정보 및 계층적인 사용자 그룹을 관리 기능을 제공한다.

#### (1) 문서종류등록 기능

이 기능은 문서등록 및 검색시에 문서종류 정보를 선택하게 함으로서 입력오류 외에 현재 등록된 문서종류만을 선택하게 함으로서 불필요한 입력을 배제할 수 있다.

#### (2) 사용자관리 기능

시스템을 사용하기 원하는 사용자를 등록하고 사용자의 등록정보를 수정/삭제하는 기능을 가진다. 또한 이 기능에서는 사용자를 해당 사용자 그룹으로 매칭시킨다.

#### (3) 그룹관리 기능

사용자그룹을 계층적으로 생성 및 삭제하도록 하는 기능을 가지며 이러한 정보는 문서 보안에 이용된다.

### 3.2.5 그외 기능들

본 시스템에서 제공하는 위에서 설명한 기능 외의 기능들로는 분류별 문서검색 기능, 자료실, 자유게시판 등이 있다.

#### (1) 분류별 문서검색

분류별 문서검색기능은 문서검색을 문서분류별로 찾아가면서 문서를 검색하는 기능이다. 또한 문서등록 정보에 대한 수정 및 문서 삭제기능을 제공한다.

#### (2) 자료실

자료실 기능은 문서를 자신의 사용자 그룹이외의 다른 사용자들과 문서를 공유하기 위하여 게시하는 기능을 제공한다.

#### (3) 자유게시판

자유게시판 기능은 사용자간 의사 교환을 목적으로 구성되었다. 이 기능은 글쓰기, 글읽기, 수정, 삭제 등으로 구성되어 있다.

## 3.3 구현환경

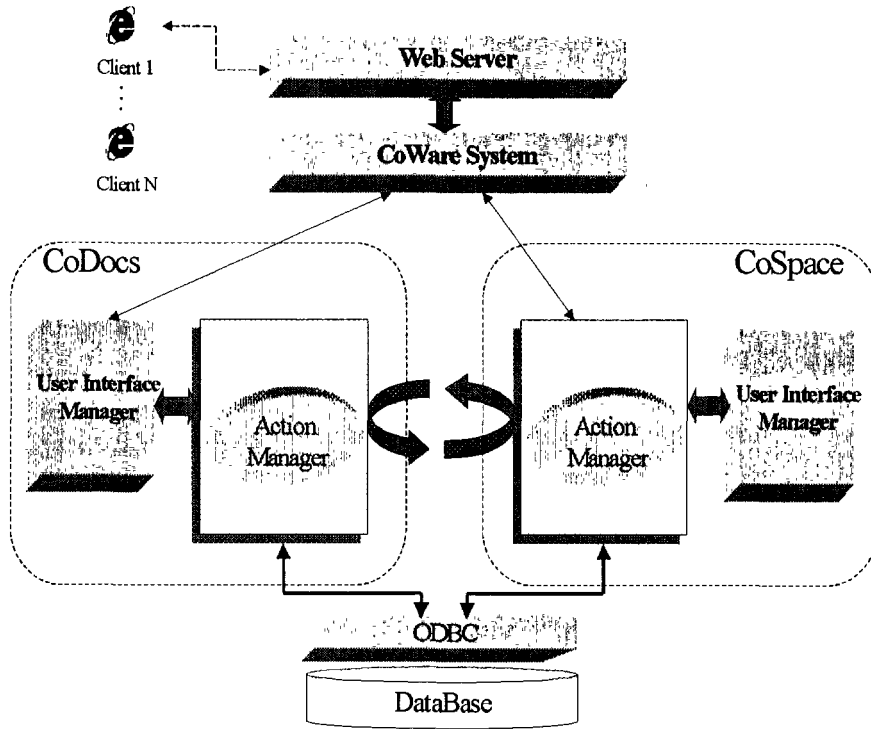
본 연구에서는 운영체제로 Window NT 서버와 마이크로소프트사의 IIS (Internet Information Server) 4.0 웹 서버를 사용하였고, 데이터베이스로는 MS SQL Server 6.5를 사용했다. 또한 Action Manager는 ODBC 인터페이스를 사용했으므로 ODBC를 지원하는 모든 데이터베이스와 연결이 가능하다[7][8]. 사용자 인터페이스 관리자(User Interface Manager)는 C언어로 구현했고, Action Manager는 C++를 사용하여 구현하였다.

## 4. 시스템 통합 및 인증모델

CoWare 시스템은 이미 구축된 기술문서 관리 시스템(CoDocs)와 공동작업 공간(CoSpace) 시스템을 통합함으로써 효율적인 공동작업을 가능하게 하였다. 여기서는 통합된시스템의 개략적인 시스템 구조와 시스템 전반에 걸친 사용자 인증에 대해 기술하고 있다.

### 4.1 CoWare 통합 시스템

본 시스템은 기존에 구현된 문서관리 시스템(CoDocs)와 공동작업 공간(CoSpace) 시스템을 통합하여 하나의 시스템으로 구축하였다. 이미 구축된 두 개의 시스템은 구조적으로 거의 비슷한 형태를 가지고 있으며 CoDocs와 CoSpace에서 사용되는 데이터베이스 정보를 하나의 조직체처럼 운용하여 어느 시스템을 사용하더라도 자료의 투명성을 유지할 수 있다. 또한 두 개의 Action Manager가 서로 소켓을 이용한 통신을 함으로써 시스템간의 자료 공유에 있어 효율적인 작업을 할 수 있게 하였다. 이러한 시스템의 구조를 그림으로 보면 다음과 같다.



[그림 4-1 CoWare 시스템 구조도]

시스템을 사용하는 사용자는 웹 브라우저를 통해 CoWare 시스템으로 접속을 하게 되면 시스템 내부적으로는 한번의 시스템 로그인 과정을 통해서 CoWare 시스템의 적법한 사용자로 인증을 받게 된다. 이미 구축되어 있는 CoDocs와 CoSpace 두 개의 시스템이 하나의 정보조직을 공유해서 쓰게 되므로 인증 작업을 마친 사용자는 두 시스템중 어느 것이라도 사용 가능하며 자료를 공유해서 사용할 수 있게 하였다.

## 4.2 통합 시스템의 인증 모델

CoWare 시스템은 기존의 CoDocs와 CoSpace 두가지 시스템을 통합하여 하나의 시스템으로 구축한 것이다. 통합된 모델에서 사용자 인증은 두 시스템이 하나의 사용자 정보 조직을 이용함으로써 한번의 인증으로 두 시스템을 사용 가능하게 하였다.

### 4.2.1 ASP 인증 모델

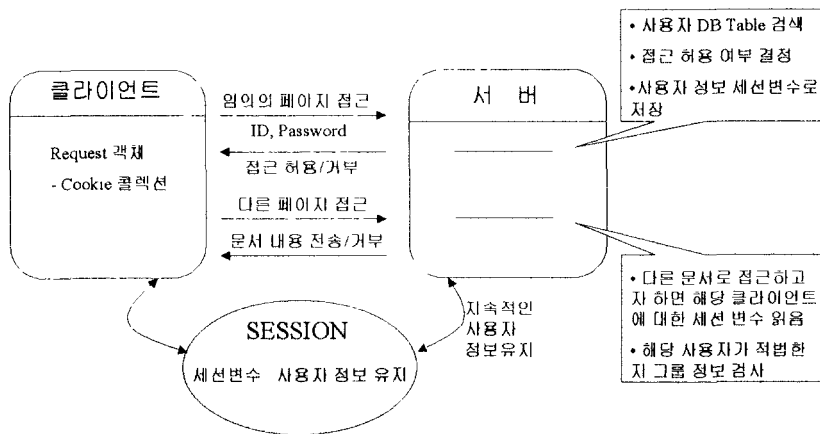
본 시스템에서는 IIS(Internet Information Server)와 연계해서 ASP(Active Server Page) 스크립트 언어를 사용하여 특정 사용자에 대한 인증 체계를 정의하고 있다[5][6]. 여기서는 ASP의 Session(세션) 객체라는 것을 이용하는데, 이는 임의의 사용자가 시스템에 접속하게되면 그 웹 브라우저가 종료될 때까지 접속한 사용자에 대한 정보를 세션객체에 저장해두고 그 정보를 유지하게 된다. 기본적으로는 이러한 세션객체에 사용자에 대한 아이디와 패스워드 등을 포함해두고 사용자 인증 및 문서에 대한 접근의 방법을 제어한다.

세션 객체라는 것은 사용자가 처음 어느 페이지에 접속했을 때 각 클라이언트는 세션

객체로 할당된다. 이러한 세션객체는 각 사용자들이 어떤 시스템에 접속한 이후 여러 페이지들을 옮겨 다닐 때 임의의 특정 정보를 지속적으로 유지시키고자 할 때 필요 한다. 이 객체는 각각 고유의 세션(Session) ID로 구분되어 지는데, 이 ID는 사용자가 사이트를 브라우징하는 동안 페이지에서 세션을 나타내는데 사용되는 것이다. 실제 이러한 ID정보는 클라이언트 쪽에서 Cookie정보로 저장되어진다. 또한 이 객체는 사용자가 마지막 요구를 보낸 후 20분(기본 지정 값)이 지나면 자동 종료되거나 사용자가 원할 때 임의로 종료될 수 있다.

세션 변수는 세션 객체에 저장된다. 이러한 세션 변수는 세션의 정보를 저장하는 데 이용된다. 또한 이 변수들은 세션에 관한 특성상 하나의 클라이언트에 대해서 글로벌 한다. 즉, 사용자가 임의의 사이트에서 브라우징 하는 동안은 지속적으로 유지되는 정보이다. 본 시스템에서 적용한 세션을 이용한 인증 방법은 위에서 언급한 세션 변수에 사용자의 정보(아이디, 패스워드, 그룹정보)를 저장해서 사용자가 임의의 문서를 참고하고자 할 때 이 정보를 읽어와서 합당한 사용자인지를 검사하게 된다.

이러한 ASP 세션(Session) 인증 방법을 그림으로 보면,



[그림 4-2 ASP 인증 모델]

위와 같은 인증 모델의 절차를 따라보면 다음과 같이 요약될 수 있다.

- (1) 보호된 페이지로 임의의 사용자가 접근을 하게 되면 서버는 사용자의 확인을 위해 아이디와 패스워드를 묻는다.
- (2) 사용자가 입력한 정보가 서버내의 정보조직에 있다면 접근을 허용하게 되고 이때 접근한 각 클라이언트에 따라 고유 세션이 생성되고 세션변수에 그 정보를 넣게 된다.
- (3) 임의의 사용자가 다른 페이지로 접속을 요구하면 서버에서는 해당 사용자의 세션 변수를 읽어와서 해당 사용자가 요구한 페이지에 대한 적절한 그룹의 사용자인지를 인증한다.

#### 4.2.2 일반화된 인증 모델

일반적인 인증 모델은 Access Control Matrix라 하여 각 객체에 대해 접근 권한을 리스트 형태로 지니고 있고 이 권한 리스트와 비교하여 접근을 허락하는 인증 방식을 취한다. 또한 여기에 응용된 형태로 여러 가지 모델이 있을 수 있는데 많이 응용되는 방식이 각 객체에 대한 계층적 조직 정보를 가지고 인증을 하고자 하는 경우이다.

##### 4.2.2.1 Access Control Model

일반적인 개념의 액세스 컨트롤은 컴퓨팅 시스템에서 폭넓게 사용되어 왔는데 현재 존재하는 많은 액세스 모델은 Access Matrix Model이라는 것에서 발전되어 왔다.

일반적으로 이러한 액세스 모델에는 보호되어야 하는 수동적 형태의 리소스에 대한 객체들의 집합체와 이러한 리소스들을 접근하고자 하는 능동적 형태의 subject들이 존재한다. 또한, 이러한 subject와 객체들과 연관되어질 수 있는 접근권한이 존재할 수 있다. 이러한 접근권한은 하나의 subject가 임의의 하나의 객체를 접근할 수 있는 권한을 부여한다.

Access Matrix로부터 발전된 이러한 구체적인 모델들은 아래와 같은 성질들을 포함한다.

(1) matrix를 column으로 나눌 수 있는 능력. 즉, 각각의 사용자들은 액세스 가능한 객체들의 리스트를 유지하고 있다. 새로운 객체에 대한 추가는 각각의 사용자들의 접근목록이 수정되어야 하는 것을 필요로 하고 있다.

(2) matrix가 row로 나누어질 수 있는 access control list. 즉, 각각의 객체들은 그것과 상호작용할 수 있는 사용자들의 리스트를 가지고 있다.

본 시스템에서는 이러한 Access Control Model과 더불어 일반화된 인증 방법의 하나로써 정의될 수 있는 조직구조의 개념의 시스템의 사용자 인증에 도입하고 있다.

##### 4.2.2.2 조직적인 구조를 가지는 인증 모델

이 모델의 경우는 각 사용자들에 대한 그룹 구성 및 해당 그룹의 접근 권한을 계층적 구조로 생각하여 권한 부여를 하는 것이다. 임의의 한 사용자는 자신 및 자신의 부모에 접근 권한이 설정된 문서만을 접근할 수 있는 경우이다. 하지만 같은 부모를 가진다고 하더라도 자신의 이웃 노드 사용자(sibling node)에 접근 권한이 부여된 문서는 접근할 수 없다.

이와 같은 권한 설정 정보들은 일반적인 text 형태로는 설정하기 어려우므로 DB table에 저장하여 부여된 권한에 대한 정보를 유지하고 원래 사용자 그룹에 대한 구성 자체가 계층적 형태이므로 부모와 자식 노드 사용자에 대한 것들은 각 DB table간에 연결 리스트 형태로 그 정보를 구성할 수도 있다.

#### 4.2.3 시스템의 인증 모델

본 시스템에서 사용되는 사용자 인증 모델은 기본적으로는 앞에서 설명한 접근 권한 모델중 조직적인 구조를 가지는 인증 모델을 사용한다. 즉, 각 사용자들은 자신이 접근할 수 있는 문서들에 대한 접근 권한 목록을 유지하게 되고 반대로 각 문서들은 자신을 접근할 수 있는 각 사용자 및 그룹에 대한 접근권한 목록을 유지하게 된다. 본 시스템에서는 이러한 Access Matrix Model을 기본적인 인증 구조로 정하고 있다. 여기에 더불어 각 사용자들은 자신이 임의의 사용자 그룹에 속해 있고 이러한 사용자 그룹들 역시 계층적인 구조로 이루어져 있기 때문에 각 사용자 그룹들 간의 권한 부여도 할 수가 있다. 즉, 시스템에

서는 정보조직이라는 사용자 및 그 사용자 그룹에 대한 계층적 구조를 유지함으로써 그 사용자가 어느 정도의 접근 권한을 가지는지를 결정할 수가 있다.

또한 ASP 스크립트 언어의 개념을 도입하여 각 사용자들의 고유 정보는 ASP의 세션 객체를 이용하여 유지하도록 한다. 이 세션은 임의의 사용자가 처음 시스템에 접근했을 때가가 사용자마다 생성되는 것으로 사용자가 시스템에서 종료하여 빠져나갔을 경우나 시스템에서 내부적으로 정해놓은 특정 시간동안 사용자가 아무런 작업을 하지 않아 세션이 종료될 때까지 그 사용자에 대한 정보를 유지하게 된다.

이러한 경우 시스템에서 일어날 수 있는 인증 시나리오를 살펴보면 다음과 같다.

- (1) 사용자가 처음에 시스템에 접근할 때
  - ㄱ. 해당 사용자가 적절한 사용자인지를 정보조직 내의 사용자 정보와 비교
  - ㄴ. 적절한 사용자라면 사용자에 대한 세션 생성
  - ㄷ. 사용자에 대한 정보를 ASP 세션 객체에 저장, 유지
- (2) 새로운 사용자를 등록할 때
  - ㄱ. 아이디, 패스워드 및 기타 사용자에 대한 정보 기록
  - ㄴ. 사용자가 소속될 그룹 및 그 그룹의 상위 그룹에 대한 정보 유지
- (3) 임의의 문서를 등록할 때
  - ㄱ. 그 문서를 접근할 수 있는 그룹 범위 설정
    - 자신 및 자신이 속한 그룹의 관리자에게 접근 허용
    - 자신이 속한 그룹의 모든 멤버에게 접근 허용
    - 자신의 그룹 및 상위 그룹의 관리자에게 접근 허용
    - 자신의 그룹 및 자신에 대한 형제 그룹 사용자에게 접근 허용
    - 작업과 관련된 다른 프로젝트 팀들에 대한 접근 허용
  - ㄴ. 등록될 문서가 포함될 문서 정보 조직에 추가
- (4) 임의의 사용자가 임의의 문서를 접근하고자 할 때
  - ㄱ. 접근하고자 하는 문서의 권한 허용에 관한 설정을 읽어옴
  - ㄴ. 접근을 원하는 사용자가 문서에 설정된 접근권한 허용 리스트에 포함되는지를 비교
  - ㄷ. 위에서 설명한 4가지 경우에 포함되는 사용자라면 접근 허용

## 5. 관련연구

공동작업 시스템은 지리적으로 분산된 컴퓨터 사용자들이 공동작업을 할 수 있도록 지원해주는 시스템으로 네트워크가 발달되면서 그 효용 가치가 더욱 커지고 있다. 현재 정보를 공유하고 정보의 활용성을 높이기 위한 공동작업시스템이 많이 개발되고 있는데 공동작업 시스템은 특화된 서비스를 요구하기 때문에 전용의 하드웨어/소프트웨어를 설치해야 하는 경우가 많다. 기존의 공동작업을 위해서는 사용자가 전용 시스템이 설치된 플랫폼이 반드시 필요하고 일반적인 컴퓨터 환경에서 공동작업을 진행하는 것은 어려웠다. 결국 특

정 플랫폼과 전용 시스템의 도입을 낳게 되었고 다른 종류의 공동작업 시스템과는 상호운용이 불가능하다. 그러므로 기존의 공동작업시스템을 이용하여 공동작업에 참가하려는 사용자들은 반드시 같은 종류의 시스템 환경을 가져야 한다. 이와 관련된 연구로서 Public Space[10], TeamRoom[11], JCE[12], MBone[13], Rendezvous[14] 등이 있다. 이러한 특정 플랫폼과 전용시스템 도입의 문제들을 해결하기 위하여 웹을 기반으로도 연구가 되어졌는데 대표적으로 GMD의 BSCW 시스템[3][4], NCSA의 HyperNews[15], Conference Mosaic[16] 등을 들 수 있다.

특히 BSCW 시스템은 웹을 기반으로 하는 공유작업공간(Shared WorkSpace)을 지원하기에 공동작업을 위한 전용 시스템 도입의 문제를 해결했고 또한 특정 플랫폼과 네트워크 환경에 종속되는 문제점들을 해결했다. 그러나, BSCW 시스템은 작업처리 속도가 느리고 그룹에 대한 정의가 모호하며 사용자가 능동적으로 그룹에 참가할 수 없는 단점이 있다. 그 뿐만 아니라 그룹단위의 공동작업을 효과적 지원해 주기 위해서는 특정한 문서들을 등록, 관리하고 추후에 검색 관리 할 수 있는 기능이 필요하다. 그러나 BSCW 시스템은 이러한 문서관리 기능을 지원해 주지 않는다.

본 연구에서는 이와 같은 문제점들을 해결하고 공유작업공간과 함께 웹 기반에서 체계적이고 효율적인 기술문서관리 기능을 제공하는 문서관리시스템을 통합함으로써 보다 효과적인 정보시스템을 구현하였다. 아래는 BSCW와 CoSpace/CoDocs의 기능을 비교한 표이다.

	BSCW	CoSpace/CoDocs
그룹종류	단일그룹	특성에 따른 세가지 그룹지원
그룹참여 형태	초청(Invite)	초청, 능동적 참여(Join)
속도	느리다	속도개선 (서버프로세스가 데이터베이스 포인터 유지)
문서관리 기능	없음	지원
메일서비스	지원	없음(예정)

또한 문서관리 시스템에 관한 연구도 여러 분야에서 진행중인데 문서관리 기능을 지원해 주는 그룹웨어로는 핸디오피스 등이 있다. 웹 기반의 연구로는 스마트플로우98, PCDOCS의 CyberDOCS가 있다. CyberDOCS는 문서검색, 문서의 버전정보를 관리하기 위한 Check Out, Check In 기능, 문서생성 기능, 인증, 히스토리(history)기능, 문서의 메타데이터 기능 등을 지원한다.

그러나 기존의 연구에서는 웹을 기반으로 공동작업을 지원해 주는 시스템과 기술문서관리를 지원해 주는 문서관리 시스템을 통합하여 지원해 주는 시스템은 없었다.

## 6. 결론

본 연구에서는 공동작업을 위한 CoSpace공동작업 시스템과 정보서비스를 지원해 주는 CoDocs문서관리 시스템을 구현하였다. CoSpace공동작업 시스템을 통해 사용자들은 자신의 공동작업과 관련되어진 그룹의 구성원들과 효과적으로 정보들을 서로 공유하고 교환할 수 있다. CoDocs문서관리 시스템의 유용한 기능들을 통해 사용자들은 중요한 기술문서들

을 체계적이고 효율적으로 관리할 수 있고 조직 내에 있는 다른 사용자들에게 기술문서들을 공유할 수 있다. 또한 두 시스템의 Action Manager 구조는 항상 데이터베이스와 연결 포인터(Connection Pointer)를 유지하고 관리함으로써 속도, 향상, 비용 절감 등등 효율성을 높일 수 있는 구조를 가지고 있다.

구현된 두 시스템은 웹을 기반으로 구현되었으므로 특정 플랫폼에 독립적이고 웹 서비스가 되는 곳이라면 어디서나 공간의 제약을 받지 않고 사용이 가능하며 공동작업에 참여하고자 하는 사용자들은 특별한 응용 프로그램 없이 웹 브라우저만 있으면 이를 통하여 공동작업에 참가 할 수 있다.

사용자들에 대한 인증은 위의 두 시스템을 통합한 통합 인증 모델을 사용하였다. 통합한 두 시스템에 등록된 임의의 사용자는 통합 시스템에 처음 접속 했을 때 사용자를 위한 세션이 만들어지고 세션이 종료될 때까지 사용자에 대한 정보를 유지하게 된다. 만들어진 세션은 사용자가 CoSpace공동작업 시스템과 CoDocs문서관리 시스템을 사용할 때 별도의 인증 과정 없이 정보를 유지하고 있는 각각의 세션을 통하여 사용자에 대한 인증이 이루어지게 된다. 그리고 공동작업 시스템과 문서관리 시스템을 통합함으로써 공동작업으로부터 생성되어진 기술, 설계문서들을 쉽게 문서관리 시스템에 등록할 수 있고 이 후에 효율적으로 문서들을 관리할 수 있다.

추후 연구로 위의 두 시스템을 통합한 CoWare 통합시스템으로 계속 발전해 나가하고자 한다. 먼저 CoSpace공동작업 시스템은 그룹을 구성하는 공동작업 구성원들간의 효율적인 작업 진행을 위해 토론장 기능을 추가하고 효과적인 문서등록을 위해 CoSpace 시스템 안에서 그룹 공동작업으로부터 생성된 기술문서들을 쉽게 CoDocs 시스템에 등록하도록 하는 기능을 추가할 것이다. 또한 CoDocs 시스템은 문서관리 시스템에 이미 등록된 기술문서들에 대해 버전관리를 함으로써 기술문서를 효과적으로 재사용 할 수 있도록 하여 기술 문서의 관계성과 연계된 지식을 관리하게 하고 적당한 조건을 만족하는 기술문서들을 백업하는 기능 및 CoDocs 시스템 전반에 걸친 백업을 지원하고자 한다. 특별히 구성원들이 상호간에 실시간 메시지 교환 기능, 메시지 notification 기능, 원하는 작업 구성원과의 작업 공유를 위한 미팅룸(Meeting Room) 기능을 향후 제공할 것이다.

## 감 사

본 연구는 “과학기술부-한국과학재단 지정 울산대학교 기계부품 및 소재 특성평가 연구센터”의 지원에 의한 것입니다.

## 참고문헌

- [1] Berners-Lee, T., Cailliau, R., Luotnen, A., Frystyck Nielsen, H. and Secret, A., The World-Wide Web, in Communications of the ACM, 37(8), August, 1994
- [2] E. Nebel, L. Masinter, "Form-base File Upload in HTML", RFC 1867, November 1995



- [3] Bentley, R., Horstmann, T., Sikkel, K. and Trevor, J., "Supporting Collaborative Information Sharing with the World Wide Web: The BSCW Shared Workspace System", in The World Wide Web Journal: 1, December 1995, ©O'Reilly
- [4] Bentley, R. and Appelt, W., "Designing a System for Cooperative Work on the World-Wide Web: Experiences with the BSCW System", in Proceedings of HICSS'30: January 1997, ©IEEE Computer Society Press
- [5] Simson Garfinkel, with Gene Spafford, "Web Security & Commerce", O'Reilly & Associates, 1997
- [6] Scot Johnson, et al "Special Edition Using Active Server Pages", QUE, December 1997
- [7] Microsoft Press, "Microsoft ODBC 3.0 programmer's Reference, Volume1, and SDK Guide", Microsoft, 1997
- [8] Microsoft Press, "Microsoft ODBC 3.0 programmer's Reference, Volume2", Microsoft, 1997
- [9] John F.Patterson, Ralph D.hill, Steven L.Rohall, "Rendezvous : An Architecture for synchronous Multi-User Application," Proc. ACM CSCW 90, ACM Press, New York, 1990
- [10] Frank Reiff, "PublicSpace: A Flexible Shared Workspace System", 11 September 1997, ECSCW'97
- [11] Roseman, M. and Greenburg, S. (1996) "TeamRooms: Groupware for Shared Electronic Spaces." In the Proceedings of Chi '96, British Columbia, Canada.
- [12] Bjørn J. Kvande, "The Java Collaborator Toolset: A Collaborator Platform for the Java Environment," 1996
- [13] H.Ericksson, "MBONE: The Multicast Backbone," Communication of the ACM, Vol20, No9, 1994
- [14] John F.Patterson, Ralph D.hill, Steven L.Rohall, "Rendezvous : An Architecture for synchronous Multi-User Application," Proc. ACM CSCW 90, ACM Press, New York, 1990
- [15] Daniel Laliberte, "What is HyperNews ? : A Brief Overview," from Internet, <http://www.hypernews.org/HyperNews/get/hypernews.html>
- [16] Euihyun Jung, Yong-jin Park, Hideyoshi Tominaga, Jaehong Lee, Choongwoog Lee, "A Design and Implementation of Conference Mosaic," Proc. of the 10th International Conference on Information Networking, Jan, 1996
- [17] 김규완, 정수권, 정재훈, 김인호, 이명준, "유연성있는 웹기반의 문서관리시스템", 한국정보과학회 '98가을 학술논문 발표집(II) 제25권 2호 p.222-224, 1998
- [18] 정재훈, 정수권, 김인호, 김규완, 이명준, "효과적인 공동작업 지원을 위한 웹기반 공유 작업공간", 한국정보과학회 '98가을 학술논문 발표집(II) 제25권 2호 p.270-272, 1998