우 종 정 교수지도

석사학위 청구논문

# FLASH를 이용한 웹 기반 문서 저작 도구의 설계 및 구현

2005

성신여자대학교 교육대학원

교육학과 전자계산교육전공

이 승 민

# FLASH를 이용한 웹 기반 문서 저작 도구의 설계 및 구현

우 종 정 교수지도

이 논문을 석사학위 논문으로 제출함

2004년 12월

성신여자대학교 교육대학원

교육학과 전자계산교육전공

이 승 민

# 인 준 서

이 승 민 의 석사학위 논문을 인준함

심사위원 (인)

심사위원 (인)

심사위원 (인)

성신여자대학교 교육대학원

# 논문 개요

인터넷 환경의 발달과 함께 다양하고 효율적인 웹 기반 문서의 개발이 요구되면서 멀티미디어문서가 더욱 활용되고 있다. 이에 따라 많은 웹 문서 저작도구시스템이 등장하였으며, 멀티미디어 문서를 저작하기 위한 프로그램들도 부가되었다. 하지만, 이러한 저작 도구는 대부분 전문 지식이 필요하며, 그에 따른 습득지식도 요구된다.

본 논문은 현재 알려져 있는 멀티미디어 프로그램인 플래시의 스크립트를 이용하여 효율적인 웹 문서 저작 도구 시스템을 제안하고자한다. 이 시스템은 웹상에서의 가벼운 플랫폼으로 동작하므로 사용자 접근이 용이할 뿐만 아니라 화려한 멀티미디어 문서 저작이 용이하고, 모든 유무선 단말기에 탑재 가능하며, 시스템 확장 및 유지 보수에 용이하다.

# 목 차

논문개요

Ⅰ. 서론 ..... 1

Ⅱ. 관련연구 ..... 3

1. 플래시와 플래시 Action Script ..... 3

2. 플래시의 특징 ..... 4

3. 저작도구 ..... 5

Ⅲ. DASF의 설계 ..... 7

1. DASF의 개발 방향 ..... 7

2. DASF의 구성 ..... 8

(1) 문서 저작 초기 단계 ..... 9

(2) 문서 저작 단계 ..... 10

(3) 문서 저장 단계 ..... 12

Ⅳ. DASF의 구현 ..... 13

1. 시스템의 구성 ..... 13

2. 문서 저작 초기 단계 ..... 13

3. 문서 저작 단계 ..... 16

(1) 저작 단계 초기화 ..... 18

(2) 객체 모음창 ..... 19

(3) 객체 속성 변경창 ..... 21

(4) XML load와 XML Parser ..... 23

4. 문서 저장 단계 ..... 24

5. 시스템 분석 ..... 26

Ⅴ. 결론 ..... 28

참고문헌

ABSTRACT

# 표 목차

<표1> 서버와 클라이언트 환경 ..... 13

<표2> 시스템 비교표 ..... 27

# 그림 목차

[그림 1] DASF의 흐름 ..... 8

[그림 2] 문서 초기 단계 흐름 ..... 9

[그림 3] 시스템 안에서의 객체모듈 구분 설계 ..... 11

[그림 4] 시스템 안에서의 정보적재모듈 구분 설계 ..... 11

[그림 5] 시스템 안에서의 정보인터페이스모듈 구분 설계 ..... 12

[그림 6] 문서 저장 단계 흐름 ..... 12

[그림 7] 사용자 선택 화면 ..... 14

[그림 8] 저장 파일 선택 화면 ..... 15

[그림 9] 디자인 서식 단계 선택 화면 ..... 15

[그림 10] 저작 초기 단계에서의 서버와 클라이언트의 동작 방식 ..... 16

[그림 11] 문서 저작 단계 화면 ..... 17

[그림 12] 저작 단계에서의 서버와 클라이언트의 동작 방식 ..... 18

[그림 13] Object 중 O_Circle 객체 초기화 ..... 19

[그림 14] Object Array의 초기화 ..... 19

[그림 15] 객체 모음창 ..... 20

[그림 16] 객체생성함수호출 기능담당 함수(call_creat())호출 ..... 20

[그림 17] call_creat() 안에서의 객체 생성모듈 호출 ..... 20

[그림 18] 생성모듈 ..... 20

[그림 19] 객체 속성 변경창 모음의 일부 ..... 21

[그림 20] 속성 변경창 안에서의 call_resize 호출 ..... 22

[그림 21] call_resize 안에서의 객체 속성 변경 모듈 호출 ..... 22

[그림 22] 너비 변경 후 Array에 저장 ..... 22

[그림 23] 깊이 변경 후 Array에 저장 ..... 22

[그림 24] XML 문서 ..... 23

[그림 25] XML load ..... 24

[그림 26] XML Parsing ..... 24

[그림 27] 저작 단계에서의 저장화면 ..... 25

[그림 28] 저장한 후의 SWF 내려받기 화면 ..... 25

[그림 29] Array를 XML로 구성 ..... 25

[그림 30] 저장 단계에서의 서버와 클라이언트의 동작 방식..... 26

# Ⅰ. 서론

최근 몇 년간 컴퓨터 성능의 향상과 멀티미디어 및 정보통신 기술의 발전으로 인터넷이 급속도로 성장하였으며 또한 웹의 보편화로 인터넷 사용자수도 기하급수적으로 늘어났다[1]. 이러한 인터넷 발전은 정보에 대한 다양한 수요와 욕구를 만들었으며, 웹 문서의 생산성 증가를 요구하게 되었다. 이에 따라 일반사용자도 쉽게 접근할 수 있고 강력한 생산성을 지닌 '저작 도구(Authoring tool)'가 출현하였다.

저작 도구는 일반 사용자가 전문 프로그램을 사용하지 않고도 웹 문서를 저작할 수 있으며, 기존 전문 프로그래밍으로 문서를 저작할 때보다 개발 시간을 단축할 수 있다. 하지만 현재 대부분 저작도구들은 DLL로 구성된 PC기반의 시스템으로써, 일반 사용자는 접근이 용이하지 않으며, HTML과 서버 스크립트로 구성된 어플리케이션 형태의 저작도구는 서버에 편중된 수행으로 인하여 원활한 서비스 제공이 용이하지 않다. 또한 이러한 시스템들은 모든 형태의 유무선 단말기(PC, PDA, Phone, E-book, etc)에서 서비스를 제공하기 어렵다. 따라서 본 시스템은 시간적·공간적 제약 없이 시스템의 접근 가능하고, 모든 클라이언트에 탑재될 수 있으며, 실시간 스트리밍으로 전송 받을 수 있고, 파일 손상 없이 일정한 화질을 이용할 수 있는, 플래시의 SWF를 이용한 웹 기반 문서 저작 시스템(Document Authoring System using Flash : DASF)을 설계 구현하고자 한다.

본 연구의 내용은 다음과 같이 구성된다. 1장은 연구의 필요성과 구성을 제시하고, 2장은 배경연구로 플래시(Flash)와 Action Script, 플래시의 특징, 저작 도구에 대하여 살펴본다. 3장은 본 연구에서 제안한 DASF에 대

한 설계의 제시 방향과, 구현을 위한 설계 및 흐름도에 대해 살펴본다. 4장은 실제 구현된 시스템의 내용을 살펴본다. 끝으로 5장은 본 연구의 결론을 제시한다.

# Ⅱ. 관련연구

## 1. 플래시와 플래시 Action Script

플래시는 미국 Macromedia사가 만든 소프트웨어로서 벡터 그래픽을 기반으로 한 동영상 제작 프로그램이다. 간단한 드로잉과 애니메이션 제작이 가능한 프로그램으로 다이나믹하고 인터렉티브한 무비파일 제작과 프리젠테이션, 포트폴리오 등의 작업을 아주 간단하고 편리하게 할 수 있는 저작툴이다[2]. 플래시 Action Script는 플래시에 내장된 언어로써, 객체 지향을 기반으로 한 스크립트언어이다. 초기 플래시는 애니메이션을 위한 간단한 스크립트언어와 함께 애니메이션 그래픽 툴로 출발하였다. 1994년 플래시 4.0에서 플래시 스크립트를 'Action Script'라 명칭하고, 기본적인 프로그램 언어에 쓰이는 부분들이 추가되었다. 플래시 5.0에서는 객체 지향 스크립트 언어와 XML소켓 지원 등 진보된 프로그램 언어로 발전하여, 클라이언트 컴퓨터를 활용하는 클라이언트측 어플리케이션을 개발할 수 있게 되었고, 플래시 MX로 접어들게 되면서 액션 스크립트의 객체개념이 보다 확실해졌고, 외부와의 데이터 교환도 강화되었다[3].

# 2. 플래시의 특징

플래시는 벡터 그래픽 기반으로, 이는 어떠한 환경에도 화질 손상 없는 이미지를 제공한다. 스트리밍 기술을 지원하기 때문에, 전체 데이터가 전송되기 전에 데이터를 재생할 수 있으며, 강력한 애니메이션 제작과 Action Script을 통한 사용자와의 상호작용 부여에 따른 UI개발을 구현할 수 있으며, SWF안에서의 데이터를 생성, 저장, 수정하여 서버와 필요한 데이터만을 주고받음으로써, 기존의 HTML과 서버측 스크립트를 기반으로 개발된 웹 어플리케이션보다 서버의 부담을 감소시킨다. 객체지향이라는 개념을 채택하여 관리 및 유지 보수에 용이하며, 확장성을 갖는다. 또한, 인터넷 PC이외에 PDA나 모바일, 셋톱박스 환경에도 용이하게 접속 할 수 있다.

플래시의 특징을 정리하면 다음과 같다[4].

- 벡터 그래픽 기반으로 일정한 화질을 유지하며, 강력한 사운드 지원과 동적인 플래시 애니메이션 제공한다.
- 일반적인 GIF 애니메이션에 비해 적은 용량으로, 압축률이 뛰어나다.
- 자연스러운 움직임의 정보를 선사하여 인터넷에 동적인 개념을 부여한다.
- 브라우저의 플러그 인을 사용하여, 스트리밍 기술을 지원한다.
- 객체지향개념의 스크립트언어이다.

# 3. 저작도구

저작도구(Authoring tool)란 자체 프로그램만을 가지고 일반적인 응용프로그램을 만들 수 있는 프로그램을 뜻한다[5]. 즉 프로그램을 만들 수 있는 프로그램이라고 할 수 있다[6,7,8]. 현재 활용되고 있는 저작 도구들의 유형을 살펴보면 다음같이 대략 5가지의 유형으로 분류할 수 있다.

- 프리젠테이션 도구(Presention Tools)

  프리젠테이션은 매력적인 디스플레이를 위해 텍스트, 그래픽, 오디오, 애니메이션, 점핑과 많은 비주얼 효과들을 포함하는 슬라이드들을 사용하여 만들 수 있다. 그리고 분기에 있어서는 사용자가 원하는 위치를 어디로든 점프할 수 있을 뿐만 아니라 다른 애플리케이션으로도 점프가 가능하고 다시 돌아올 수도 있다. 이러한 도구는 마이크로소프트사의 파워포인트, 아도브의 퍼스웨이젼, 로터스의 프리랜스 등이 있다.

- 페이지 기반의 저작도구(Page Based Authoring Tools)

  페이지 기반은 전통적인 책과 같은 모조의 페이지처럼 표현할 수 있다. 페이지 내용을 장별로 나눌 수 있고, 메뉴를 색인할 수 있으며, 어떤 페이지로 점프할 수 있다. 인덱스 기능, 탐색 기능, 등을 위해서 버튼을 구성하며 텍스트, 그래픽, 오디오, 비디오, 애니메이션, 분기와 같은 많은 비주얼 효과를 포함할 수 있다.

- 프레임 기반의 도구(Frame based Tool)

  애플리케이션 안에 있는 각 디스플레이를 프레임이라 부르며, 텍스트, 그래픽, 애니메이션, 비디오, 오디오와 같은 넓은 영역의 멀티미디어 구성요소들로 프레임의 내용을 만들 수 있으며, 라이브러리로부터 단순히 만들어진 템플릿으로도 구성할 수 있다. 그리고 템플릿은 내용 프리젠테이션과 상호작용을 제공하며 완전한 스크린 그래픽 백그라운드 라이브

러리도 이용 가능하다.

아이콘 기반의 도구(Icon Based Tool)

아이콘 방식은 프로그램을 설계할 때 플로우챠트식으로 프로그램을 만드는 방법으로 일련의 연속되는 흐름을 자유롭게 조정할 수 있는 저작 방식으로서, 응용프로그램상의 각 프레임들을 손쉽게 연결할 수 있다[1].

시간을 기반으로 하는 도구 (Time-Based Tool)

각각의 요소 및 이벤트들에 일정한 시간간격을 부여하여 한 프로그램 안에서 사용자가 지정한 어떤 이벤트 부분으로 편리하게 건너뛸 수 있도록 사용자와 프로그램 간의 인터렉티브 기능을 강화한 프로그램방식이다[1].

# Ⅲ. DASF의 설계

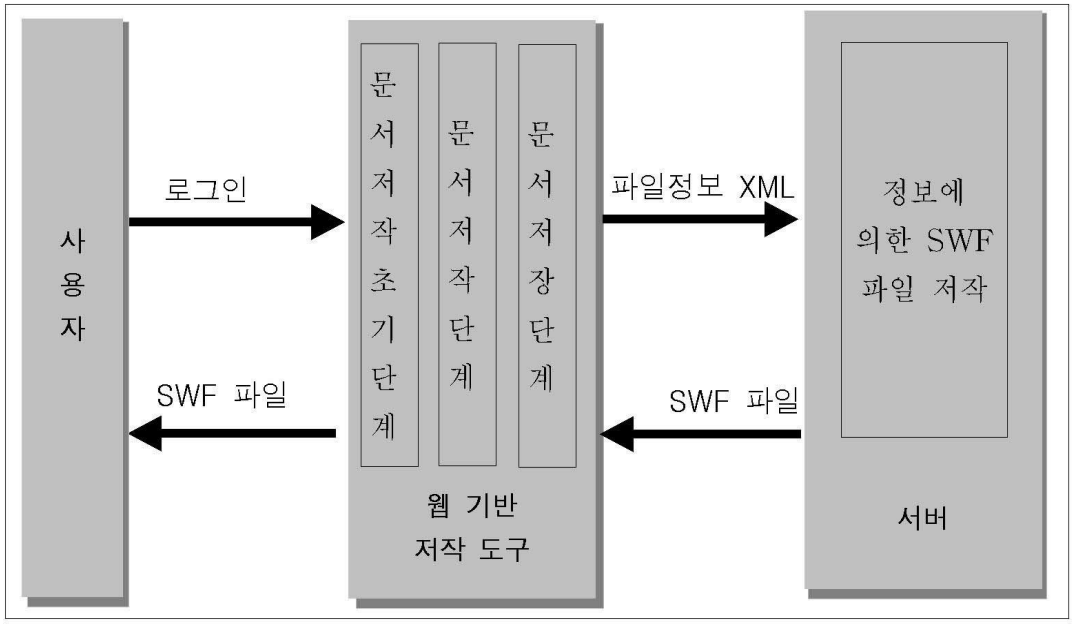
## 1. DASF의 개발 방향

DASF는 웹 기반 저작 도구 시스템으로 다음과 같은 개발방향을 설정한다.

- 접근 용이성
  기존의 저작 도구는 일반 사용자가 직접 구매하고, 클라이언트에 설치하여야 하며, 사용 장소에 제한을 받는다. 이러한 단점을 보완하고자 시·공간적 제약에서 벗어난 웹 기반의 문서 저작 시스템을 이용하여 문서 저작, 저장, 편집, 내려받기 받을 수 있도록 한다.
- 유무선 환경을 위한 시스템
  웹 기반 문서 저작 도구는 네트워크 컴퓨터 환경에 적합하기 위해, 모든 클라이언트와 유무선 단말기에 탑재될 수 있는 가벼운 시스템이여야 하고, 화면 크기에 따른 파일 손상 없이 일정한 화질을 유지해야 한다. SWF는 압축률이 뛰어나 가벼운 시스템으로 적합하며, 벡터 방식을 이용하기 때문에 화질 유지가 용이하다.
- 뛰어난 멀티미디어 제공
  Flash의 멀티미디어 저작 도구라는 장점을 이용하여 멀티미디어 문서를 표현하기 위한 이미지, 사운드 애니메이션 등이 풍부하게 제공될 수 있도록 한다.
- 사용 및 유지보수의 편리성
  시스템의 흐름을 설정하여 사용 편리성을 구성하고, 후에 시스템의 관리 및 유지보수의 편리성을 위하여 객체지향모듈로 구성하며, 시스템을 부분적으로 나누는 것을 제시하고자 한다.

# 2. DASF의 구성

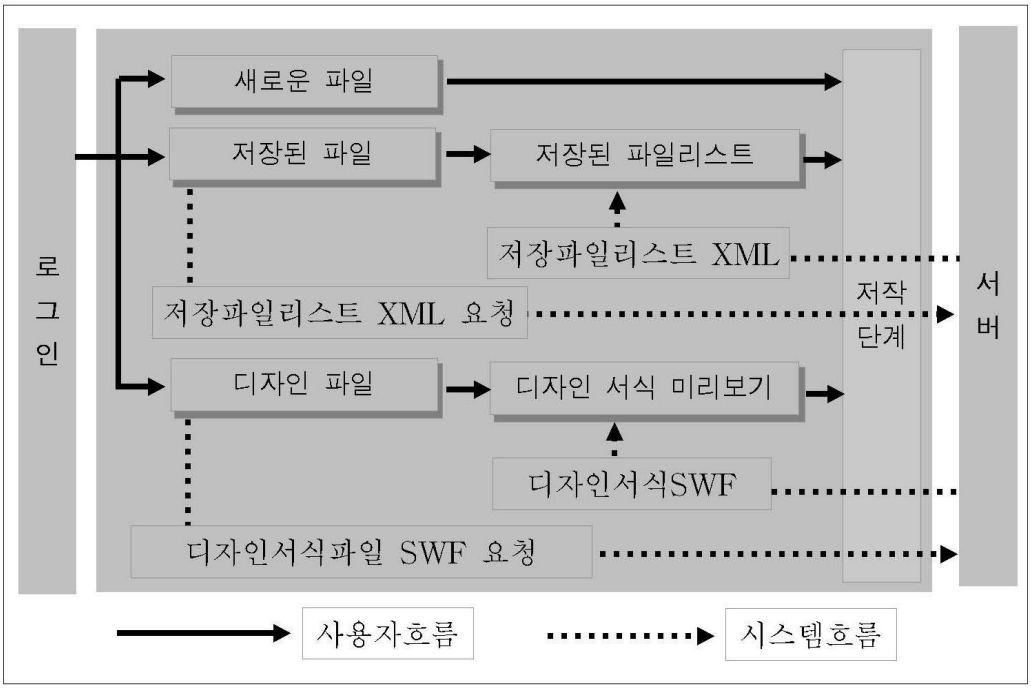
DASF는 사용자의 편리성 향상을 위해 [그림 1]과 같이 시스템 흐름에 따른 동작 방식으로 구성하였다. 사용자는 문서 저작 시스템에 로그인하여 문서 저작 초기 단계에서 새로운 파일이나 저장 파일, 디자인 서식 파일을 선택하여, 문서 저작 단계에서 문서를 저작 및 편집할 수 있고, XML 형태로 서버에 저장할 수 있으며 끝으로 문서 저장 단계에서 저작된 문서 정보를 통해 서버가 SWF 문서를 생성하면, 사용자는 이 SWF 문서를 내려받기 할 수 있도록 구성하였다.




[그림 1 DASF의 흐름]

(1) 문서 저작 초기 단계

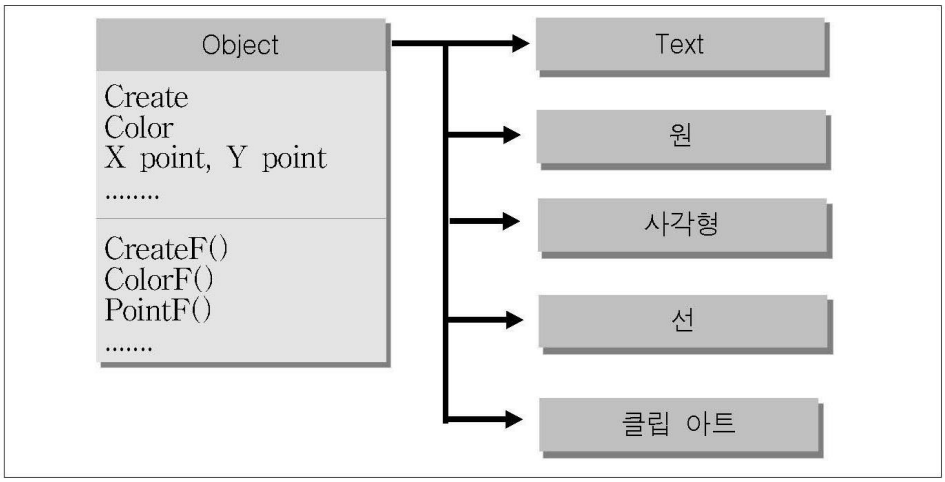
문서 저작 초기 단계는 문서 저작 전 파일 선택의 단계로써, 고려할 사항은 다음과 같다. 새로운 문서 저작을 기본으로 하며, 사용자에게 파일편집 용이성을 제공하기 위하여 저장 파일 정보 제공과 문서 저작의 다양한 욕구를 만족 시켜줄 디자인 서식 제공이 필요하다. 이에 따라 사용자는 로그인을 한 후, 새로운 파일, 저장된 파일, 디자인 서식 파일을 선택할 수 있도록 구성한다. [그림 2]과 같이 새로운 파일을 선택한 사용자는 바로 다음 단계인 문서 저작 단계로 이동할 수 있도록 하고, 저장 파일을 선택한 사용자에게는 시스템 서버에 저장된 저장 파일리스트와 파일미리보기창을 제공하도록 한다. 디자인 서식을 선택한 사용자에게는 서버에서 제공하는 디자인 서식파일을 미리보기할 수 있도록 구성한다. 저장파일리스트는 클라이언트가 서버에 리스트를 요청하면, 서버는 다 량의 구조적 데이터 교환을 가장 용이하게 표현할 수 있는 XML로 리스트를 작성하여 클라이언트에게 전달한다.



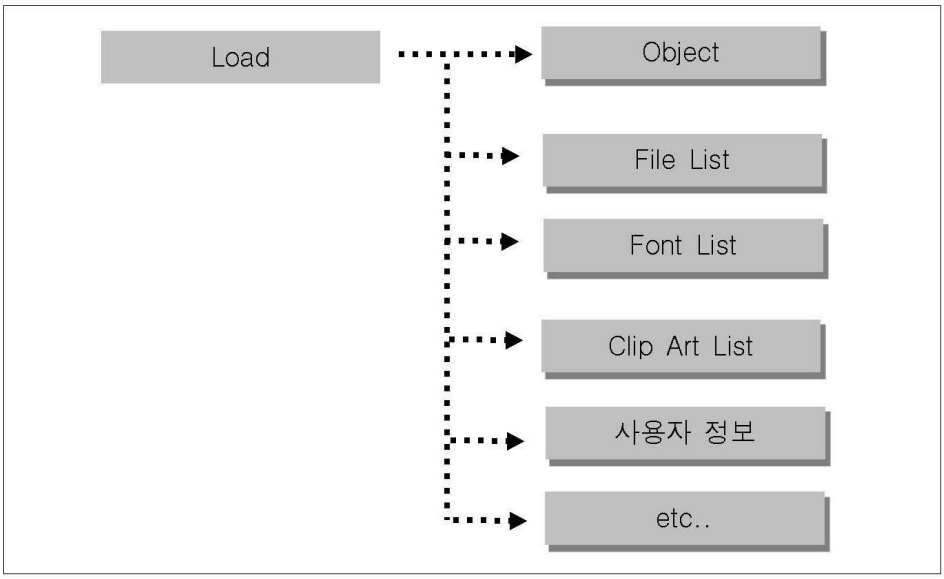

[그림 2] 문서 초기 단계 흐름

(2) 문서 저작 단계

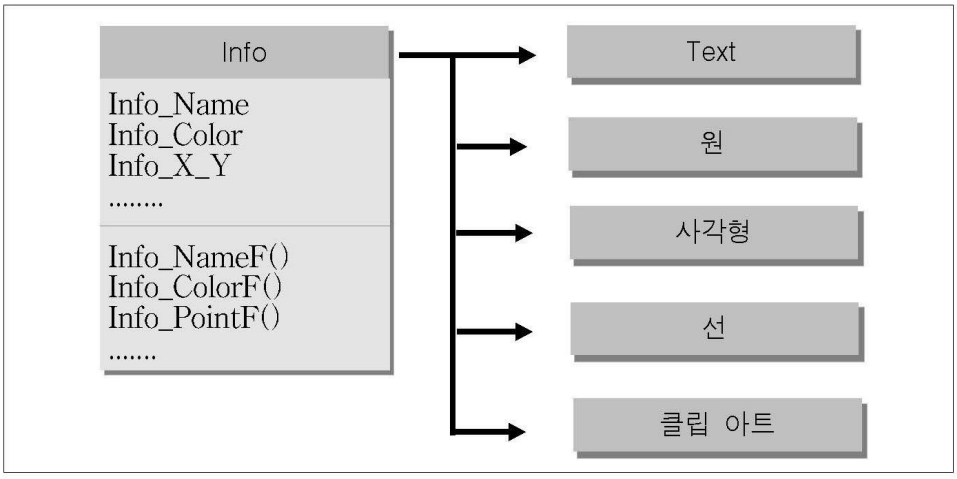
문서 저작 단계는 문서를 저작할 수 있는 단계로 고려할 사항은 다음과 같다. DASF 시스템은 웹을 통한 시스템 수행이므로 시스템 자체의 수행 능력을 높이기 위한 모듈을 필요로 한다. 또한 사용자 문서 저작 시 필요한 기본적인 객체가 구성되어야 하고, 사용자 인터페이스의 편리성을 가져야 한다. 또한 멀티미디어 문서 저작을 위한 기능이 필요하다. 시스템 자체의 수행 능력 향상을 위해서는 객체간의 관계성 및 속성과 상속성을 이용하여 객체 클래스 모델을 설정하고 클래스를 그룹화 한다. 객체 모델링은 시스템 자체의 수행 능력을 높힐 수 있을 뿐만 아니라, 모듈의 관리 및 유지 보수가 수월하며, 재활용성이 뛰어나다. 또한 시스템의 접근성을 위하여 시스템 전체를 적재하는 방식이 아닌 사용자 이용에 따른 부분 시스템 적재 방법을 사용하도록 한다. DASF는 객체모듈, 정보적재모듈, 정보인터페이스모듈, XML Parser 모듈과 같이 다수의 모듈로 구성된다. 객체모듈은 시스템 안에서 모든 객체를 담당하는 모듈로써, [그림 3]와 같이 객체속성에 의해 세분화된다. 정보적재모듈은 서버와 클라이언트 사이에 이동하는 정보를 적재하는 기능을 제공하며 [그림 4]와 같이 정보에 따라 모듈이 구분된다. 정보인터페이스모듈은 객체에 관한 속성정보와 사용자가 내리는 명령 하달 기능을 담당하는 모듈이며 [그림 5]과 같이 정보인터페이스 모듈은 객체에 따라 구분된다. DASF의 클라이언트와 서버간의 정보이동은 XML을 이용하며, XML Parser 모듈이 담당한다. 시스템에서 텍스트, 이미지, 클립아트는 기본으로 사용자에게 제공되며, 좀더 강력한 멀티미디어 기능을 포함하는 문서 저작을 위하여, 텍스트 이펙트, 애니메이션 클립아트, 그림 툴, 사운드, 배경 이미지를 제공한다. 사용자가 작성중인 문서 정보는 객체 Array에 저장되며, XML 언어형식으로 변환되어 서버로 이동된 후, 저장된다.




[그림 3] 시스템 안에서의 객체모듈 구분 설계



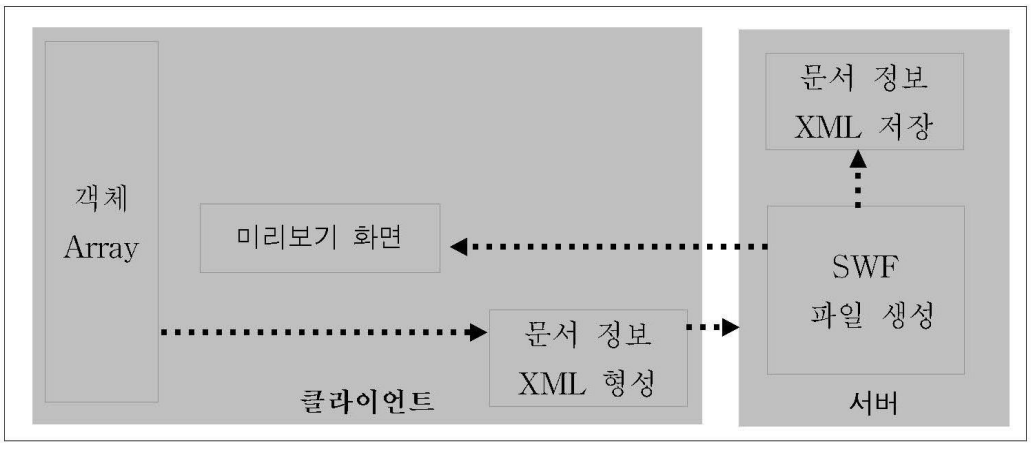

[그림 4] 시스템 안에서의 정보적재모듈 구분 설계




[그림 5] 시스템 안에서의 정보인터페이스모듈 구분 설계

(3) 문서 저장 단계

문서 저장 단계는 저작한 문서 파일을 저장하는 단계로, 문서 저작 내용을 미리 보기 할 수 있어야 하며, 문서 파일을 서버에 저장할 수 있도록 하고, 파일을 다운 받을 수 있도록 한다. [그림 6]과 같이 사용자가 저작한 문서 정보가 담겨있는 객체 Array를 XML로 형성하여 서버에 보내고, 서버는 XML의 정보를 이용하여 SWF 문서를 생성하여 미리보기 화면을 구성하고 XML 파일을 서버에 저장하도록 한다.




[그림 6] 문서 저장 단계 흐름

# Ⅳ. DASF의 구현

## 1. 시스템의 구성

웹 기반 문서 저작 시스템의 실행 및 구현 환경은 다음과 같다.
시스템 구현 서버는 WINDOWS 2000 SERVER이며 IIS를 사용하였고, 사용자 정보를 담기 위한 Database로는 MYSQL를 사용하였다. 서버와 클라이언트 사이에 정보들은 XML를 사용하여 정보 교환을 한다. 사용자 인터페이스는 ASP와 Flash를 이용하였다.

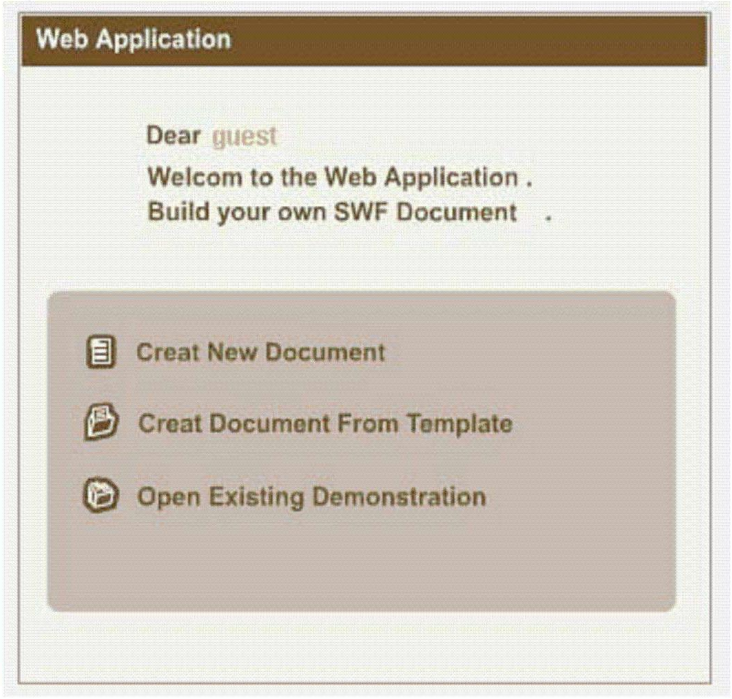
<표 1> 서버와 클라이언트 환경

| | | | |
|---|---|---|---|
| 서버 | Flash 5.0 | 클라이언트 | Flash Player |
| | IIS & MySQL | | ASP & HTML |
| | WINDOWS 2000 SERVER | | MS Explorer 5.0 이상 |
| | | | WINDOWS 2000 |

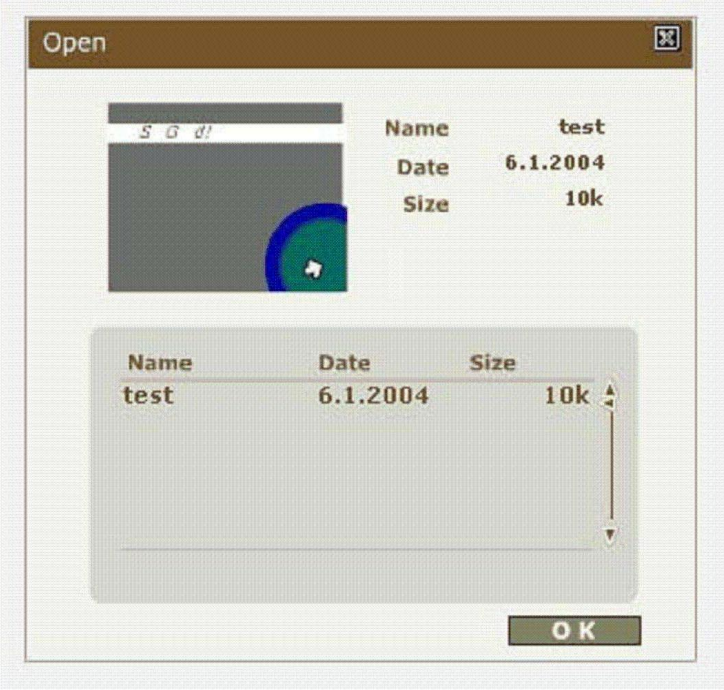
## 2. 문서 저작 초기 단계

DASF는 사용자 로그인 후 [그림 7]과 같이 문서 초기 단계를 접할 수 있다. 문서 초기 단계는 새로운 파일, 저장 파일, 디자인 서식 파일을 선택

할 수 있도록 구성하였다. 사용자가 새로운 파일을 선택하면, 다음 단계인 문서 저작 단계를 이용할 수 있으며, 저장 파일를 선택하면 [그림 8]과 같이 저장 파일리스트와 미리보기 창으로 이동하고, 디자인 서식파일를 선택하면 [그림 9]와 같이 디자인 서식 미리보기 창으로 이동할 수 있도록 구성하였다. 사용자가 저장파일을 선택하였을 때, 클라이언트가 서버에게 사용자의 저장 파일리스트 XML을 요청하며 서버는 클라이언트가 요청한 저장 파일리스트 XML을 클라이언트에게 전달한다. 서버로부터 XML을 받은 클라이언트는 XML Parser 모듈을 통하여 XML를 Parsing하여 객체 Array에 저장하고 사용자에게 저장 파일리스트와 미리보기화면을 제공한다. [그림 10]은 저작 초기 단계에서의 서버와 클라이언트의 동작 방식이다.

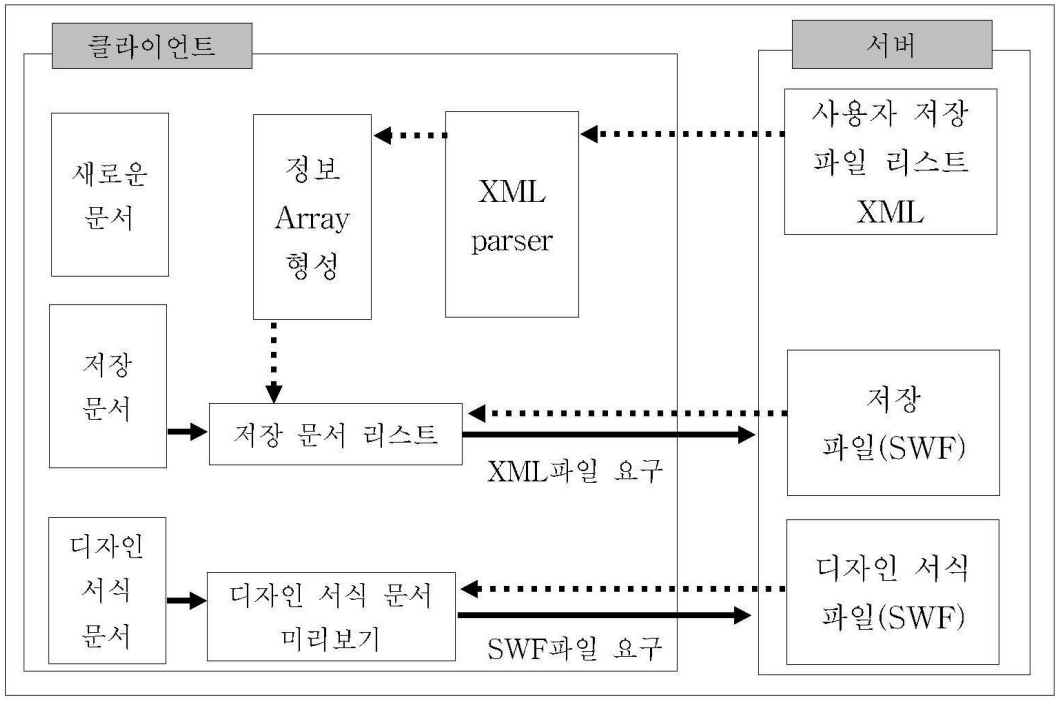



[그림 7] 사용자 선택 화면




[그림 8] 저장 파일 선택 화면

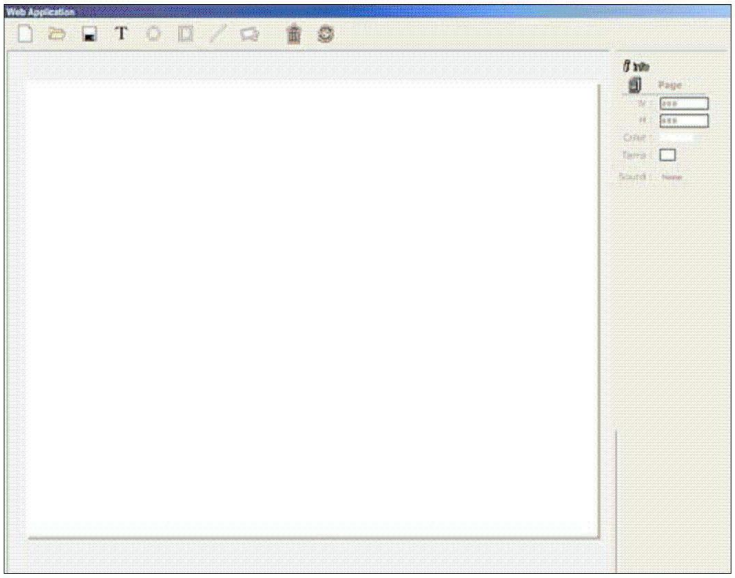
[그림 9] 디자인 서식 단계 선택 화면




[그림 10] 저작 초기 단계에서의 서버와 클라이언트의 동작 방식

# 3. 문서 저작 단계

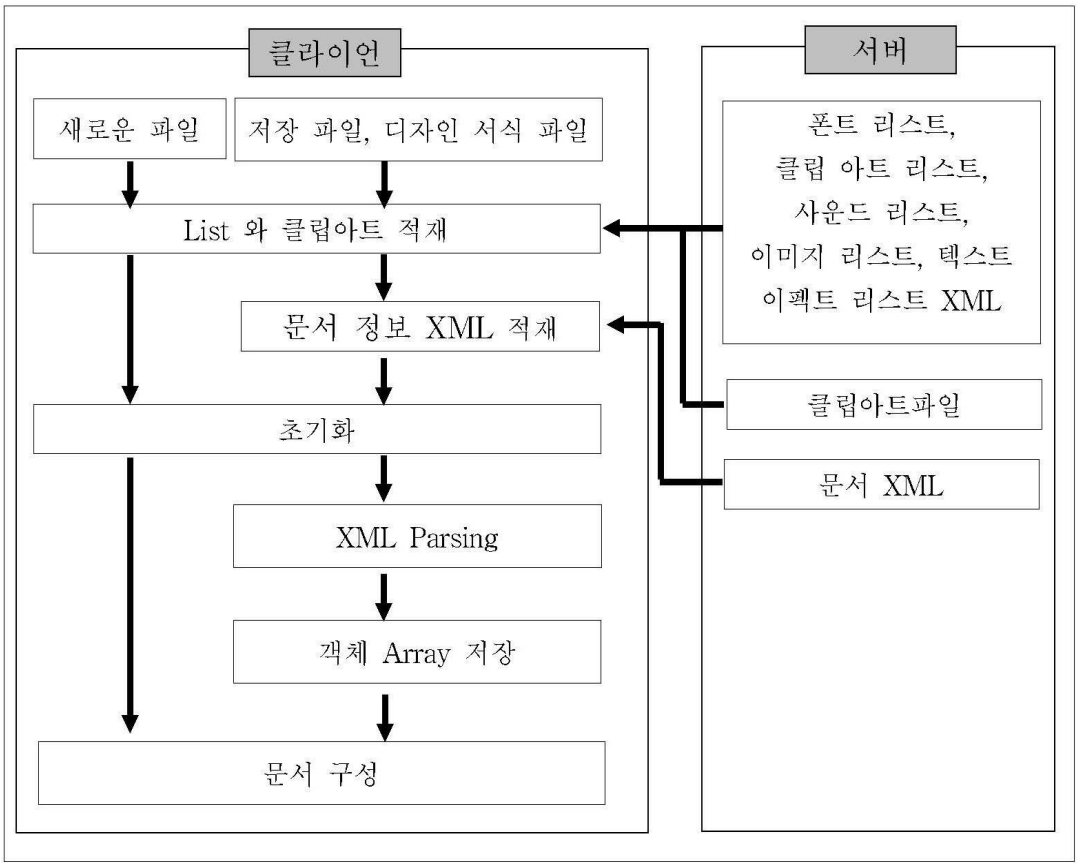
문서 저작 단계는 기본 텍스트를 이용한 일반 텍스트 문서뿐만 아니라 도형 그리기, 클립아트, 사운드, 텍스트 이펙트, 애니메이션 클립아트 등을 이용하여 멀티미디어 문서를 저작할 수 있도록 구성하였다. [그림 11]은 문서 저작 단계의 화면이다.



[그림 11] 문서 저작 단계 화면

화면 상단에는 사용편리성을 위하여 객체모음창을 구성하였으며, 객체모음창은 클릭을 통하여 객체를 생성할 수 있다. 화면 오른쪽 객체정보창은 선택된 객체정보를 나열하며, 객체의 속성을 변경시킬 수 있다. 객체는 각기 고유의 구분자를 갖고, 이 구분자는 객체생성과 편집에 관한 모듈 호출 시 사용된다. 저작 단계는 전 단계인 저작 초기 단계에서 새로운 파일 선택, 저장 파일 선택, 디자인 서식 파일 선택과 같이 세 가지 경로로 접근하게 된다. 첫 번째 경우인 새로운 파일 선택 후의 저작 단계에서는 인터페이스에 필요한 각종 리스트 즉, 폰트 리스트나 클립아트 리스트, 클립아트 파일 등을 적재한다. 그 후 저작 단계 초기화 모듈이 실행된다.

저장 파일이나 디자인 서식 파일 선택 후에는 새로운 파일 선택 시와 초기 부분은 같으며, 인터페이스에서 필요한 리스트들의 적재시 문서 정보 XML도 함께 적재한다. 저작 단계 초기화 후 문서 정보 XML을 Parsing하여 객체 Array에 저장한 뒤 객체의 구분자를 보고 그에 맞는 생성모듈을 호출하여 객체를 생성하며, 객체 Array에 저장되어 있는 객체속성 정보를 보고 객체속성모듈을 호출, 생성된 객체를 변형하여, 문서에 구성한다. [그림 12]은 저작 단계에서의 서버와 클라이언트의 동작 방식이다.




[그림 12] 저작 단계에서의 서버와 클라이언트의 동작 방식

(1) 저작 단계 초기화

저작 단계 초기화 모듈은 객체의 활성 및 객체 속성 정의, 객체 초기화를 수행하고, 객체 Array를 생성하며, 변수 등을 초기화 하는 기능을 담당한다. 예를 들면 [그림13]와 같이 'new Object' 메쏘드를 이용하여 객체를 활성하고 '__proto__'를 이용하여 속성함수를 정의한다. 그리고 [그림14]와 같이 'new Array' 메쏘드를 이용하여 객체 Array를 생성한다. 객체 Array는 클라이언트안에 존재하며, 서버로 문서 정보를 보내기 전까지 사용자가 저작한 문서 정보를 담고 있다.

```
O_Circle = new Object;
O_Rec = new Object;
O_Text = new Object;
O_Clipart = new Object;
__proto__.creat = Objectcreat;
__proto__.locate = location;
__proto__.Wsize = ObjectWsize;
__proto__.Hsize = ObjectHsize;
__proto__.rotate= rotation
```

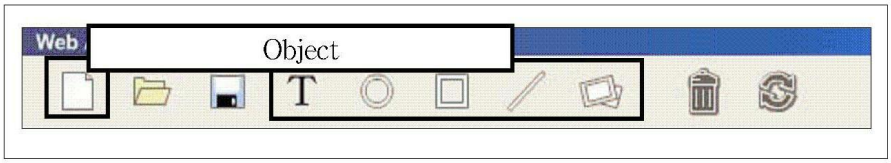
[그림 13] Object 중 O_Circle 객체 초기화

```
// ***Clip 에 관한 정보
CStyle = new Array();   // format (0=jpg, 1=bmp, 2=swf)
CPath = new Array();    // pass
CPosX = new Array();
CPosY = new Array();
CSWFNum = new Array();
CDepth = new Array();  // Layer val
//Drawing 관련 정보
DType = new Array();   // form option (0=Line, 2=Rectangle, 3=Circle)
DPosX = new Array();
DPosY = new Array();
```

[그림 14] Object Array의 초기화

(2) 객체 모음창

객체 모음창은 객체생성에 대한 명령을 하달한다. 사용자의 버튼클릭과 동시에 객체 구분자와 함께 객체생성모듈호출 기능을 담당하는 'call_creat' 함수를 호출하고, 'call_creat' 함수는 객체 구분자에 따른 객체 생성 모듈을 호출한다. 호출된 생성 모듈은 [그림 18]과 같이 객체를 생성함과 동시에 기본 속성을 적용시켜 문서를 구성하고, 객체 Array에 저장한다.




[그림 15] 객체 모음창

```
on (press) {
      ..........
      _root.call_creat(this.section);
      ..........
}
```

[그림 16] 객체생성함수호출 기능담당 함수(call_creat())호출

```
function call_creat(val){
      ..........
      _root[val].creat();// 객체생성모듈호출
      ..........
}
```

[그림 17] call_creat() 안에서의 객체 생성모듈 호출

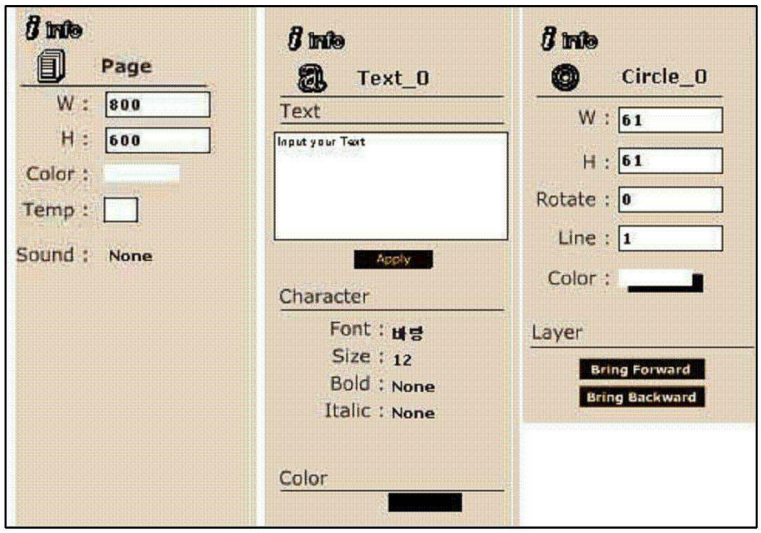
```
      _root.attachMovie( "Obj", rName, depth)
      _root[rName].attachMovie( obj, "obj",2)

      _root[rName].Classid = this.id
      _root[rName].type = obj
      _root[rName].depth = depth
      _root[rName].ANum = ArrayNum.addcount();
      _root[rName].group = false
```

[그림 18] 생성모듈

(3) 객체 속성 변경창

속성 변경창은 객체의 정보를 보여줄 뿐만 아니라 속성 변경에 대한 명령을 하달하는 기능을 담당한다. 객체를 선택할 때 객체구분자와 함께 'Info_object' 함수를 호출하고, 'Info_Object'는 객체구분자를 보고 그에 맞는 객체 속성 변경창을 제공한다. [그림 19]는 객체 속성 변경창에 대한 모음의 일부이다.




[그림 19] 객체 속성 변경창 모음의 일부

사용자가 객체 속성 변경창을 통하여 객체 속성 변경 값을 입력하였을 경우, 'call_resize' 함수를 호출하며, 'call_resize'는 객체구분자를 통하여 객체 속성 변경 모듈을 호출한다. [그림 22]과 [그림 23]처럼 호출된 속성변경모듈은 전달받은 속성 변경 값을 이용하여 객체 속성을 변경한 후 객체 Array에 값을 저장한다.

```
.....
_root.call_resize(this.section,W,H,Col,Al);
// Object size change
.....
```

[그림 20] 속성 변경창 안에서의 call_resize 호출

```
function call_resize(val,W,H,Col,Al){
    .....
    _root[val].Wsize(W);// Object width size change
    _root[val].Hsize(H);// Object height size change
    .....
}
```

[그림 21] call_resize 안에서의 객체 속성 변경 모듈 호출

```
function Wsize(W){
    .....
    this._width = W; // Object width size change
    .....
    _root.DWidth[ArrayNum] = W
    // array store
}
```

[그림 22] 너비 변경 후 Array에 저장

```
function Hsize(H){
    .....
    this._height = H; // Object height size change
    .....
    _root.DHeight[ArrayNum] = H
    // array store
}
```

[그림 23] 깊이 변경 후 Array에 저장

(4) XML load와 XML Parser

DASF 시스템은 서버와 클라이언트 사이의 정보 교환은 XML로 이루어진다. 기본적으로 XML 적재 모듈은 클라이언트 Flash에 존재하며, ASP는 Flash와 서버를 연결하는 다리 기능을 한다. [그림 24]는 XML로 표현된 문서정보의 일부분이다.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Web>
<page Width="800" Height="600">
<text FSize="10" FItalic="i" FBold="b" FText="test" FColor="22" EEff1="3"
EEff2="2" EEff3="8" EKerning="2" ELineSpace="3"
    EPosX="200" EPosY="200" EAlign="2" EOrder="2" EDepth="1"/>
<draw DType="2" DPosX="394" DPosY="372" DScaleX="100" DScaleY="100"
    DAlpha="100" DRotate="0" DThick="1" DWidth="515" DHeight="23"
    DDepth="4" DBackColor="13421619" DLineColor="13421619"/>
<clip CNum="1" CStyle="2" CPosX="355" CPosY="373" CWidth="100"
    CHeight="100" CScaleX="56" CScaleY="52" CAlpha="100" CRotate="90"
    CDepth="3" CEffect="0" CSWFNum="7"/>
</web>
```

[그림 24] XML 문서

임의의 문서 정보를 클라이언트가 요청하면 서버는 문서의 정보를 XML형식으로 전달한다. 클라이언트는 이를 [그림 25]와 같이 적재한 후 [그림 26]와 같이 XML Parser모듈을 이용하여 XML을 Parsing한다. Parsing된 정보는 객체 Array에 저장하고, 저장된 정보 중 객체 구분자를 보고 객체 생성과 속성 변경호출 기능을 담당하는 모듈을 호출하며 호출된 모듈들은 객체를 생성하고 속성을 입력하여 문서를 구성한다.

```
xmlOBJ= new XML();
xmlOBJ.onLoad = initViewer;
if (this._url.indexOf("http") == -1) {
    configURL = XMLname
    sysLog.addEntry("Retrieving system config locally.", 2);
} else if (this._url.indexOf("moock.org") != -1) {
    configURL = "210.125.96.79";
    sysLog.addEntry("Retrieving system config from:/n"+configURL, 2);
} else {
    configURL = XMLname
    sysLog.addEntry("Retrieving system config from:/n"+configURL, 2);
}
xmlOBJ.load(configURL);
```

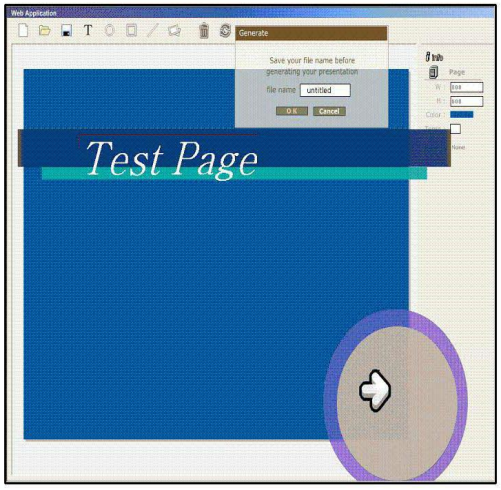
[그림 25] XML load

```
function ParseDoc (val) {
.....
for (var i=0;i<xmlOBJ.childNodes.length;i++) {
    _root.DHeight[i] = xmlOBJ.childNodes[i].attributes.DHeight;
}
for (var i=0;i<xmlOBJ.childNodes.length;i++) {
    _root.DWidth[i] = xmlOBJ.childNodes[i].attributes.DWidth;
}
}
```

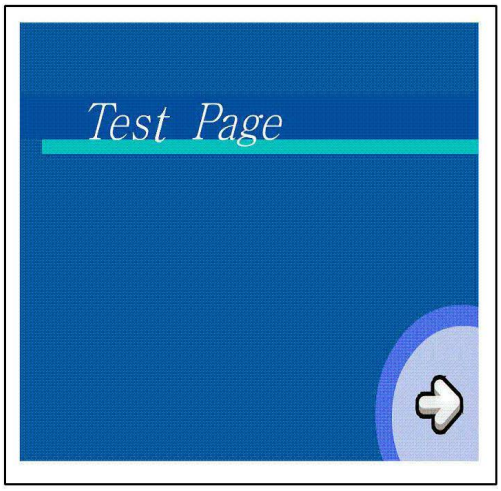
[그림 26] XML Parsing

# 4. 문서 저장 단계

문서 저장 단계에서는 저작한 문서를 서버에 저장할 수 있는 기능을 제공한다. 이때 사용자는 저작한 문서를 내려받기 할 수 있다. [그림 27]은 저장 단계에서의 화면이고, [그림 28]은 저장 단계후의 내려받기 화면이다.




[그림 27] 저작 단계에서의 저장화면

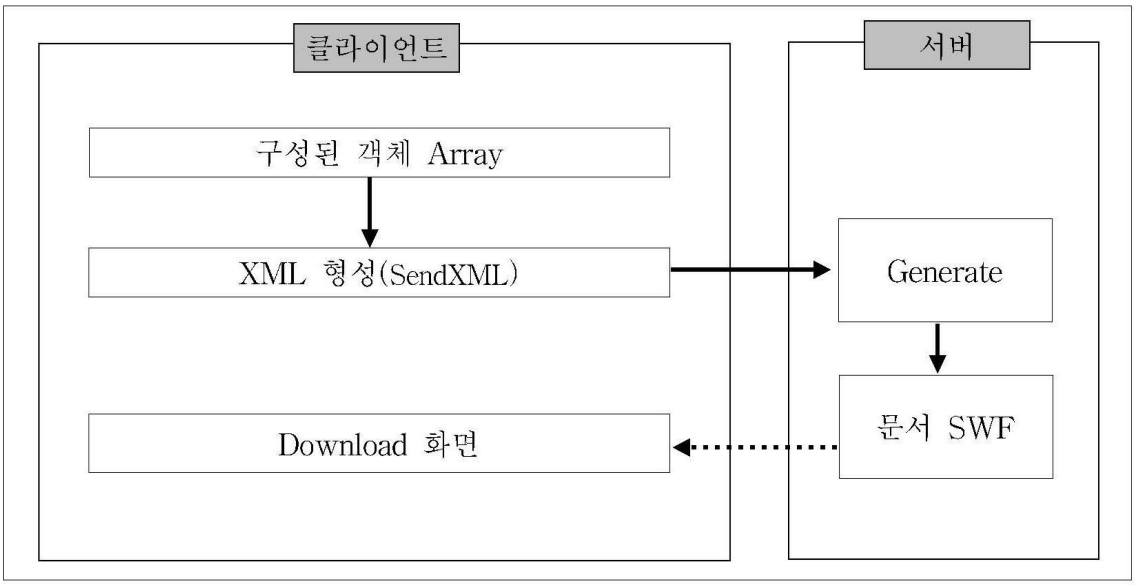



[그림 28] 저장한 후의 SWF 내려받기 화면

사용자가 문서 저작한 후 '생성버튼(Generate)'을 클릭하면 '생성버튼'은 'SendXML' 함수를 호출한다. 'SendXML'은 저장된 객체 Array들을 [그림 29]와 같이 XML로 구성하여 서버로 보낸다. 서버는 XML을 다시 Parsing 하여 그 정보에 맞는 SWF를 생성하여 [그림 28]와 같이 사용자에게 제공한다. [그림 30]은 저장 단계에서의 서버와 클라이언트의 동작 방식이다.

```
function SendXML () {
strXMLsource = "<?xml version="1.0" encoding="ISO-8859-1"?>\n";
strXMLsource += "<Web>\n";
strXMLsource += "  <page Width="800" Height="600">\n";
strXMLsource += "</Web>";
myData = new XML ();
myData.ignoreWhite = true;
myData.parseXML (strXMLsource);
.....
for (var i=0;i<_root.DWidth.length;i++) {
     myData.firstChild.childNodes[i].attributes.DWidth = _root.DWidth[i]
}
}
```

[그림 29] Array를 XML로 구성




[그림 30] 저장 단계에서의 서버와 클라이언트의 동작 방식

# 5. 시스템 분석

다음은 문서 저작 시스템에 대한 본 시스템과 기존 문서 저작 시스템을 비교한 내용이다. 본 시스템의 특징을 살펴보면 다음과 같다.

- 웹에서의 접근성이 용이하다.

  기존 저작 도구들은 시스템을 클라이언트에 설치하여 접근하지만, 본 시스템은 인터넷이 가능한 곳이라면, 시·공간 제약성이 없이 간단한 Flash Player 설치 후, 저작 도구 시스템에 접근하여 문서를 저작할 수 있으며, 저작한 문서를 다운받아 사용할 수 있다.
- 모든 형태의 유무선 단말기에 서비스 되며, 기본적인 PC뿐만 아니라 PDA등 화면 크기에 따른 파일 손상 없이 일정한 화질을 유지한다.

  본 연구시스템은 벡터 이미지인 SWF로 인터페이스와 결과물을 구성하기 때문에 가벼운 플랫폼으로 좋은 화질 유지가 용이하다.
- 일반 프로그램보다 특별한 사용방법의 숙지가 필요 없다.

기존 저작도구는 아이콘 방식의 인터페이스지만 추가적으로 HTML을 익혀야 하거나, 오픈 스크립트를 익혀야 한다. 본 시스템은 간단한 인터페이스와 도구의 이모티콘 버튼화로 시스템을 구현하였으며, 사용자에게 친숙한 Windows OS에서의 Drag&Drop 방법을 인터페이스에 가미하여, 사용 편리성을 향상시켰다.

- 멀티미디어 문서 저작이 용이하다.

  사용자가 멀티미디어 문서 저작을 위한 특별 지식이 필요하지 않도록, 다양한 애니메이션 클립아트, 텍스트 이펙트, 배경사운드, 도형 그리기 등을 제공한다.

<표 2> 시스템 비교표

| 평가요소 | 웹문서저작도구 | 파워포인트 | 툴북 | DASF |
|---|---|---|---|---|
| 접근 용이성 | 설치된 클라이언트로 접근 | | | 인터넷 접근 |
| 문서 사이즈 조정 시 화질 유지 | 문서 사이즈 재조정 필요 | | | 벡터 방식으로 재조정 필요하지 않음 |
| 사용 난이도 | 중간 | 쉬움 | 중간 | 쉬움 |
| 멀티미디어 | 멀티미디어 객체삽입가능 관련 프로그램을 익혀야 함 | 멀티미디어 객체삽입가능. 관련 프로그램을 익혀야 함 | 멀티미디어 객체삽입가능. 관련 프로그램을 익혀야 함 | 디자인 서식을 통한 멀티미디어 객체 제공 |

# V. 결론

인터넷 발전에 따른 저작도구의 출현은 일반 사용자들도 전문 프로그래밍을 하지 않고 웹 문서를 저작할 수 있게 되었다. 하지만 기존 저작 도구는 사용자 접근이 용이하지 않고 여러 형태의 유무선 단말기에 서비스하기 어렵다. DASF는 가볍고 사용하기 용이한 시스템을 기본으로 웹 공간에서 멀티미디어 툴로만 알려진 플래시의 액션 스크립트를 이용하여 구축한 멀티미디어 객체가 풍부한 문서 저작 도구이다.

본 연구에서 구현한 웹 기반 문서 저작 도구의 효과는 다음과 같다. 사용자는 시간적·공간적 제약을 벗어나 시스템에 접근 용이하며, 전문적 지식 없이 웹문서 저작이 가능하다. 플래시 SWF를 이용한 인터페이스와 SWF 문서 저작 결과물은 벡타 방식을 사용하기 때문에 기존 데스크 탑뿐만 아니라 모든 형태의 유무선 단말기 (PC, PDA, Phone, E-book, etc)에서 접근하기 용이할 뿐만 아니라, 화질 유지에 용이하며, 실시간 스트리밍 방식으로 전송 받을 수 있다. 또한 사용자는 플래시 애니메이션 특징을 지닌 화려한 문서를 저작할 수 있다. 그리고 아이콘을 이용한 인터페이스는 사용자에게 특별한 습득 시간 없이 빠르게 문서를 저작 할 수 있도록 도와준다. 시스템은 플래시 Action Script을 이용하기 때문에, 클라이언트 안에서의 데이터 생성, 저장, 수정하여 서버와 필요한 데이터만을 주고 받음으로써 서버의 부담을 줄일 수 있다. 또한 스크립트의 객체 속성을 이용함으로써, 시스템의 수행 능력을 향상시켰고 차후에 시스템 확장성이 용이하며, 관리 및 유지보수가 용이하다.

현재 많은 저작 도구들이 제작되고 있으나, 아직 기존 HTML과 서버 스크립트를 이용하거나 또는 DLL을 이용한 저작 도구들이 많다. 사용자가

다양한 경로와 다양한 제작방법을 이용함으로써 보다 쉽고 빠르게 원하는 문서 저작을 할 수 있도록 여러 종류의 저작 도구들이 개발되어야 하겠다.

# 참고문헌

[1] 김수훈, 김성환, 김성동, 최기호, “XML기반의 멀티미디어 메일 저작도구의 설계 및 구현”, ‘99년 한국멀티미디어학회 추계학술발표논문집 pp445

[2] 김동일, "국악교육 학습을 위한 웹 기반의 CAI 설계 및 구현", 2001

[3] 장일호, “애니메이션에 날개를 다는 플래시MX &액션 스크립트”, “영진.com”

[4] 나일주, “웹 기반 교육”, “과학사”, “서울”, 1999

[5] 정권순, 권오성, “멀티미디어 저작도구의 비교 및 분석”

[6] 김경숙 “CAI 개발을 위한 저작 시스템에 관한 연구” , 전국학교컴퓨터교육연구회 2권 1호 1995

[7] 이태욱, “컴퓨터교육론” “도서출판 좋은소프트”, 1999

[8] 멀티미디어교육지원센터, “멀티미디어 코스웨어 개발요원 연수교재”, 멀티미디어교육지원센터 1997

# ABSTRACT

## Design and Implementation of the Document Authoring System using Flash

Lee, Seung Min
Major in Computer Science Education
Graduate School of Education
Sungshin Women's University

The multimedia document has had a lot of attention largely due to its requirement of developing various and efficient web-base documents with growth of the internet environment.

Many web authoring tools are appeared and various functions are added to them in order to make the multimedia documents author easy. However existing authoring tools are difficult to use and mostly need export knowledge.

For these reasons, this study uses Flash scripts, which are the program to use for authoring multimedia documents, and proposes an effective system as an authoring tool. It makes the user access easy while it operates on the light platform, and it is possible to load in both wire and wireless systems, and easy make system expansion easy.