

# 웹 기반 멀티미디어 공동 작업 환경에서의 동기화 오류 제어 시스템

고 응 남<sup>†</sup>

요 약

본 연구에서는 ESS\_WMCE를 제안한다. 이 논문은 ESS\_WMCE상에서 실행되는 EDSS의 설계와 구축을 기술한다. EDSS는 웹 기반 멀티미디어 공동 작업 환경에서의 동기화 오류 제어 시스템이다. 훅킹 방법에 의하여 오류를 감지하고 오류 전달 기법은 응용 공유 시스템을 이용한 수정된 모델이다. 두레는 컴퓨터 지원 협동 작업을 위한 응용을 개발하기 위하여 지원되는 좋은 프레임워크 모델이다. 이 시스템은 기본적인 서비스 기능을 가지고 있고 기능들은 객체 지향 개념으로 구현되어 있다. 이것은 소프트웨어 기술을 사용하여 웹 기반 멀티미디어 공동 작업 환경에서 발생하는 소프트웨어 오류를 신속하게 감지하고 공유시키는 데 적합하다. 본 논문은 공유되는 객체를 접근하기 위하여 URL 동기화 뿐만 아니라 오류를 공유할 수 있다. 오류 발생시에 오류 감지는 MS-Windows API에 있는 훅킹(hooking) 방법을 이용한다. 감지된 오류는 공유되는 객체를 접근하기 위하여 오류공유가 가능하다.

## A Synchronization Error Control System for Web based Multimedia Collaboration Environment

Eung-Nam Ko<sup>†</sup>

ABSTRACT

We propose ESS\_WMCE. This paper explains the design and implementation of the EDSS running on ESS\_WMCE. EDSS is a synchronization error control system for web based multimedia collaboration environment. We have an error detection approach by using hooking method. The technique of an error transmission is a mended model of utilizing an application sharing system. DOORAE is a good framework model for supporting development on application for computer supported cooperated works. It has primitive service functions. Service functions are implemented with an object oriented concept. It is a system that is suitable for detecting and sharing a software error rapidly occurring on web based multimedia collaboration environment by using software techniques. It is able to share an error as well as providing URL synchronization to access shared objects. When an error occurs, this system detects an error by using hooking methods in MS-Windows API(Application Program Interface)function . If an error is found, it is able to provide an error sharing to access shared objects.

**키워드:** ESS\_WMCE, 웹 기반 멀티미디어 공동 작업 환경(Web Based Multimedia Collaboration Environment), URL 동기화(URL Synchronization), 오류 공유(Error Sharing)

### 1. 서 론

공동 작업은 다수의 작업 참여자 간의 의사 소통과 정보의 교환 및 공유가 원활히 이루어지도록 하여 공조 활동을 돕는 컴퓨터 기술을 총칭한다[1, 2]. 현재 컴퓨터 지원 공동 작업은 그룹 회의, 공동 설계, 공동 저작, 전자 결재 등 다양한 분야에서 연구되고 있고, 공간적으로 분산된 사용자들이 공동 작업을 하기 위해서 메일, 채팅, 화이트보드, 화상 회의 등 여러 가지 공동 작업 지원 시스템을 필요로 한다. 현재 컴퓨터 지원 공동 작업은 그룹 회의, 공동 설계, 공동 저작, 전자 결재 등 다양한 분야에서 연구되고 있고,

공간적으로 분산된 사용자들이 공동 작업을 하기 위해서 메일, 채팅, 화이트보드, 화상 회의 등 여러 가지 공동 작업 지원 시스템을 필요로 한다.

최근에는 이러한 공동 작업 지원 응용 프로그램이나 웹 브라우저 자체를 공유하는 연구가 활발하게 진행되고 있다. 공동 브라우저를 지원하기 위해서는 웹 브라우저가 제공하는 주소 지정, 스크롤, 크기, 폼 입력 등의 다양한 기능에 대하여 같은 세션의 여러 사용자에게 동기화해서 보여 주어야 한다[2-5].

본 논문의 작업 환경은 분산 멀티미디어 공동 작업 시스템에서 웹 노트라는 기능을 갖는다. 즉, 기존의 화이트보드 기능을 이용하여 응용을 인터넷 기반의 환경에서 작업 가능하도록 하는 기능이다. 인터넷의 성장과 함께 그 응용 프

<sup>†</sup> 정 회 원 : 천안대학교 정보통신학부 교수  
논문접수 : 2003년 10월 27일, 심사완료 : 2004년 1월 29일

로그래밍의 사용이 증가하고 있으며, 네트워크 대형화, 복잡화 되고 이를 이용한 업무처리가 증가하면서 네트워크 상에 많은 트래픽을 유발하게 되고 이로 인한 여러 장애 요인을 발생시키고 있다. 이는 자연히 네트워크 관리의 중요성을 부각 시키게 되었고 효율적인 네트워크 관리를 위한 여러 가지 노력들이 진행되고 있다[6]. 이러한 현재의 방향에도 불구하고 상호작용하는 멀티미디어 환경의 구성 요소에서는 그 시스템에서 계산될 수 있는 결합 허용 응용에서조차도 충분한 신뢰성(reliability)을 항상 보장하는 것은 아니다[7-9].

따라서, 본 연구에서는 웹 URL 동기화 기능을 갖는 멀티미디어 공동작업 환경에서 웹 응용 소프트웨어와 미디어 등의 프로세스(process) 요소들에 대한 오류를 미리 감지하여 알려주고 오류 동기화 시스템을 이용하여 신속하고 정확하게 전달하여서 복구하는데 빠르게 대처할 수 있는 시스템을 제안한다. 즉, 응용 공유 기능을 갖는 분산 복제형 시스템에 대한 오류를 검출하여 오류를 빠르게 공유 시키는 시스템을 제안한다.

본 논문의 구성은 2장에서는 관련 연구를 기술하고, 3장에서는 웹 기반멀티미디어 공동 작업 환경에서의 동기화 오류 제어 시스템에 대해서 기술하고, 4장에서는 시스템 평가, 5장에서는 결론을 기술한다.

## 2. 관련 연구

본 절에서는 기존 결합 허용 기법, 기존의 멀티미디어 공동 작업에 관련된 프레임워크의 종류 및 한계점 등에 대해서 기술한다.

### 2.1 기존 결합 허용 기법

결합 허용 시스템이란 하드웨어 오동작, 소프트웨어 오류 또는 정보 오염이 일어날지라도 주어진 임무를 올바르게 수행할 수 있는 시스템을 말한다[10]. 결합 허용성을 부여하는 방법에 따라 3가지로 나눌 수 있다. 첫째, 소프트웨어 기법은 운영체제에 의해 이루어지는 기법으로 소프트웨어에 의한 오버헤드로 시스템 성능 하락에 대한 희생이 따른다. 둘째, 하드웨어 기법은 하드웨어 다중화를 통해 결합 탐지 및 복구가 수행되는 기법이다. 셋째, 혼합 기법은 하드웨어로 결합을 탐지하고 소프트웨어로 결합 복구를 하게 함으로써 소프트웨어 오버헤드와 하드웨어 비용을 줄일 수 있는 장점이 있다[11]. 소프트웨어 결합허용 기법은 소프트웨어 모듈의 중복이나 재수행(rollback and retry), 또는 이 두 가지 방식의 혼용에 기초를 두고 있다. 검사점(Check pointing)은 소프트웨어 실행 중에 검사시점을 설정하여 오류가 발생했는지를 검사하여 이상이 없으면 계속 수행하고 이상이 감지되면 그 이전의 검사시점으로 되돌아가 재수행 하는 방식이다. 복구 블록(Recovery block)은 재수행(rollback and retry)에 근거한 기법으로 검사시점에서 오류가 감지되면 지정된 이전 시점으로 되돌아가 같은 기능을 가진 다른

소프트웨어 모듈을 실행하는 방식으로 단일 프로세스내에 적용될 수 있다. 분산 복구 블록(Distributed recovery block)은 복구 블록을 분산 환경으로 확장 적용하여 하드웨어 결합과 소프트웨어 결합을 동일한 방법으로 극복할 수 있도록 하는 기법이다. N-버전 프로그래밍(version programming)은 하드웨어 결합허용 기법 중 TMR(Triple Modular Redundancy)과 유사한 기법으로 N개의 독립적인 소프트웨어 모듈이 수행한 결과를 비교하여 다수의 동일한 결과를 채택하는 기법이다. TMR은 세 개의 동일한 모듈에서 실행 결과를 받아 이들의 값을 서로 비교하여 세 개 중에 두 개 이상의 결과가 같으면 이를 출력으로 발생시키는 보팅(voting) 작동에 기초한다[12]. 이외에도 목적인 응용 분야에 적합하도록 설계된 여러 가지 형태의 소프트웨어 결합 허용 기법이 있다[13, 14].

### 2.2 기존 멀티미디어 프레임워크

기존의 멀티미디어 공동 작업 환경의 종류 및 구조에 대해서 기술한다. Shastra는 Purdue 대학교에서 개발된 UNIX를 기반으로 멀티미디어 협력 작업 설계 환경을 제공하는 시스템이다. 이 시스템은 상호 작용 과정의 모든 동작을 중앙 세션 관리기를 통하여 하기 때문에 서버의 부담이 많아진다는 단점이 있다[15]. MERMAID는 일본의 Kansai C&C Lab과 NEC사에서 개발된 분산형 응용 공유 구조를 선택하면서, 공유 이벤트의 분배를 이벤트 발송 부분에서 처리함으로써 다양한 응용의 지원을 고려하는 시스템이다[16]. MmConf는 미국의 캠브리지에서 개발된 분산형 응용 공유 구조를 선택하였으며, X-윈도우즈를 기반으로 설계되어 있다[17]. CECED는 SRI international에서 개발된 중앙 집중형 구조와 복제형 구조의 혼합 구조를 지원하며, 화면 공유 개념을 확장하였다[18].

### 2.3 기존 구조에서 결합 허용 기법의 한계점

기존의 멀티미디어 공동 작업 환경의 대표적인 종류, 구조 및 제한점은 <표 1>과 같다[15-18].

<표 1> 기존의 멀티미디어 공동 작업 환경의 종류 및 개략적인 기능 비교

	Shastra	MERMAID	Mmconf	CECED
구조	분산형	분산형	집중형 또는 복제형	복제형
응용공유	지원 안됨	지원 안됨	지원 안됨	단일 응용
오류감지 (혹킹이용)	지원 안됨	지원 안됨	지원 안됨	지원 안됨
오류공유 (응용공유 이용)	지원 안됨	지원 안됨	지원 안됨	지원 안됨
웹 노트	지원 안됨	지원 안됨	지원 안됨	지원 안됨
통신 프로토콜	TCP/IP	TCP/IP	TCP/IP (multi-cast)	TCP/IP (multi-cast)

응용 공유는 구조에 따라 집중형(Centralized), 분산형(Dis-

tributed), 복제형(Replicated)으로 구분할 수 있다. 집중형 구조는 모드 구성 요소가 하나의 워크스테이션에서만 실행되는 구조이다. 이는 모든 요소가 하나의 워크스테이션에서만 실행되기 때문에 가장 간단한 방법이다. 분산형 구조는 구성 요소가 분산되어 있으며, 이들은 여러 워크스테이션에 걸쳐서 존재하는 구조이다. 복제형 구조는 분산형 구조의 변형된 형태로서, 대응하는 구성 요소가 아주 동일하거나 복제된다. 이는 각자의 워크스테이션에 실행에 필요한 모든 요소가 존재하는 구조이다. 또한, 기존 멀티미디어 공동 작업시스템에서는 응용 공유, 웹 노트라는 기능을 갖고 있지 않다[19]. 웹 노트라는 기능은 기존의 화이트보드 기능을 이용하여 응용을 인터넷 기반의 환경에서 작업 가능하도록 하는 기능이다. 특히, 기존 결합 허용 기법은 URL 동기화 기능을 갖는 멀티미디어 컴퓨터 지원 공동 작업 환경을 위한 응용 공유 구조에서의 오류 감지, 전달 등의 방법이 지원되지 않고 있다[8]. 따라서 이러한 문제점을 해결하기 위하여 본 논문에서는 사용자들 간의 공통된 웹 기반 문서를 참조할 수 있도록 웹 노트 서버의 개념을 도입시키고 이를 통해서 웹 주소(URL) 정보를 일치시킨다. 즉, 세션에 참가한 어떤 사용자와 다른 사용자가 모두 같은 웹주소 정보를 가지도록 한다. 그러나, 기존 멀티미디어 컴퓨터 지원 공동 작업 프레임워크에서는 웹 노트 기능이 없고, 웹 노트 기능을 갖는 멀티미디어 공동작업 환경에서 웹 페이지와 미디어 등의 프로세스(process) 요소들에 대한 오류를 미리 감지하여 알려주고 오류 공유 시스템을 이용하여 신속하고 정확하게 전달하여서 복구하는데 빠르게 대처할 수 있는 오류 동기화 시스템 기능이 없다.

### 3. 웹 기반 멀티미디어 공동 작업 환경에서의 동기화오류 제어 시스템

본 절에서는 웹 기반 멀티미디어 공동 작업 환경에서의 동기화 오류 제어 시스템인 ESS\_WMCE에 대해서 기술한다. URL 동기화 기능을 갖고 있는 웹 기반 멀티미디어 공동 작업 환경의 구조를 기술하고 이 환경에서의 오류 공유를 제안한다. 분산 복제형 시스템에서 혹 킹 기술을 이용하여 웹 페이지에 대한 오류를 감지하여 오류 전달 시에 웹 기반 오류 공유 시스템을 이용하여 오류를 신속히 전달하고 공유 시켜서 시스템을 신속히 안정시킨다.

#### 3.1 정의 및 표기

웹 기반 멀티미디어 공동 작업 환경을 ESS\_WMCE 라고 표시하면 이 시스템에 대한 설명과 분석을 위해서 필요한 정의 및 표기는 다음과 같다.

#### [정의 1]

웹 기반 멀티미디어 공동 작업 환경을 ESS\_WMCE 표시하면  $ESS\_WMCE = \langle P, L, M \rangle$ 이다.

여기서  $P = \{p_1, p_2, \dots, p_n\}$ 이며 프로세스(process)들의 유한 집합(finite set)이다.

$L \subseteq P^2$ 이며 채널(channel)들의 부분 집합이다.

$L = \{ \langle p_i, p_j \rangle \mid p_i : \text{메시지 보내는 프로세스}, p_j : \text{메시지 받는 프로세스} \}$

M은 메시지들의 유한 집합이다.

$M = \{ m \langle p_i, p_j \rangle \mid p_i : \text{메시지 보내는 프로세스}, p_j : \text{메시지 받는 프로세스} \}$

#### [정의 2]

웹에서의 세션이 진행 중일 때 실행되는 프로세스를 P라고 정의하기로 한다. 모든 프로세스들의 집합인 P를 공집합이 아닌 임의의 집합이라고 하면, 집합 P의 분할(partitions)  $\pi P$ 는 다음과 같은 집합이다.

$\pi P = \{SM, DM, MS, SS, AP, EDSS, A\}$ 이고

$P = SM \cup DM \cup MS \cup SS \cup AP \cup EDSS \cup A$ 이다.

$SM = \{P_{sm} \mid P_{sm} \in P, sm \in N, N : \text{자연수} \}$

$DM = \{P_{dm} \mid P_{dm} \in P, dm \in N, N : \text{자연수} \}$

$MS = \{P_{ms} \mid P_{ms} \in P, ms \in N, N : \text{자연수} \}$

$SS = \{P_{ss} \mid P_{ss} \in P, ss \in N, N : \text{자연수} \}$

$AP = \{P_{ap} \mid P_{ap} \in P, ap \in N, N : \text{자연수} \}$

$EDSS = \{P_{edss} \mid P_{edss} \in P, edss \in N, N : \text{자연수} \}$

$A = \{P_a \mid P_a \in P, a \in N, N : \text{자연수} \}$  이고

$SM \cap DM \cap MS \cap SS \cap AP \cap EDSS \cap A = \emptyset$ 이다.

여기서 SM : 세션 관리기들의 집합

DM : 데몬 들의 집합

MS : 미디어 서버 관련 집합

SS : 웹 URL 및 오류 공유 서버 관련 집합

AP : 응용 공유 들의 집합,

EDSS : 오류 감지 및 공유 시스템들의 집합

A : 응용 프로그램 들의 집합이라고 정의한다.

#### [정의 3]

본 논문에서 오류 감지 및 복구 시스템에 관련되어 있는 에이전트들의 집합은 다음과 같다.

세션이 개설되어 있을 때 여러 플랫폼(platform) 중에서 i번째 플랫폼에 실행하는 오류/URL 변경감지, 오류/URL 변경 분류와 오류/URL 공유 프로세스들을 EDSS<sub>i</sub>라고 정의한다. 정의된 에이전트들 EDSS<sub>i</sub>, EUD<sub>i</sub> 및 EUS<sub>i</sub> 사이의 관계는 다음과 같다.

분할  $\pi EDSS_i = \{ EUD_i, EUS_i \}$ 이고

$EDSS_i = EUD_i \cup EUS_i (i \in N)$ 이다.

#### [정의 4]

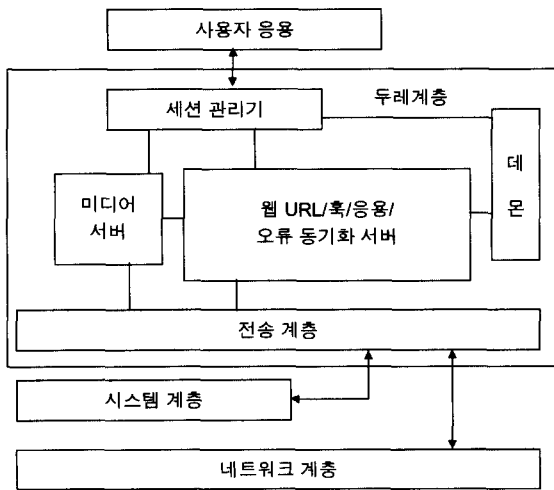
$E_i(j)$ 는 프로세스  $p_i$ 가 실행하고 있을 때 그 프로세스  $p_i$ 에서 j번째 발견되는 오류(error)들의 집합으로 정의한다.

즉,  $E_i(j) = \{e_i(j) \mid i \in N, j \in N\}$ 이다.

$S_i(j)$ 는 프로세서  $p_i$ 가 실행하고 있을 때 그 프로세서  $p_i$ 에서  $j$ 번째 발견되는 공유화된 오류(error)들의 집합으로 정의한다. 즉,  $S_i(j) = \{s_i(j) \mid i \in N, j \in N\}$ 이다.

### 3.2 ESS\_WMCE의 환경

ESS\_WMCE의 환경은 4계층으로 구성되어 있다. 시스템 계층으로는 윈도우 98/NT/2000/XP 등이 사용된다. 네트워크 계층은 분산 처리 환경의 메시지 전송을 담당하는 계층으로 MS-WINDOWS 95/98/NT의 소켓 시스템(socket system)을 활용한다. 소켓 함수의 패밀리(family)로는 AF\_INET를, 유형은 연결형을 사용한다. 즉, 네트워크 계층은 그룹통신을 지원하기 위한 방법인 TCP/IP를 이용하고 전송계층의 프로그램 지원으로 그룹 통신을 지원하는 방법과, 멀티캐스트를 이용하는 방법이 있을 수 있다. 본 제안 모델에서는 IP계층에서 호스트에 제공하는 멀티캐스트를 이용하였다. 응용 계층은 다양한 응용들이 존재하는 계층으로서 사용자들이 필요로 하는 서비스에 대한 선택이 이루어지는 계층이다. 두레 계층은 상호 참여형 멀티미디어 일반적인 응용을 개발하기 위해서 설계된 프레임워크이다. 기존의 단일 멀티미디어 응용 개발에 발생하는 미디어 제어와 세션 관리에 대한 개발 과정의 비용을 줄임으로 상호 참여형 멀티미디어 응용의 개발을 가능하게 지원하는 미들웨어이다. 본 시스템의 환경은 (그림 1)과 같다. 본 논문에서 제안하는 시스템은 여러 기능의 에이전트가 존재하며 원활한 오류 감지 및 복구 기법을 수행하는 멀티 에이전트 시스템인 EDSS 이다.



(그림 1) ESS\_WMCE의 환경

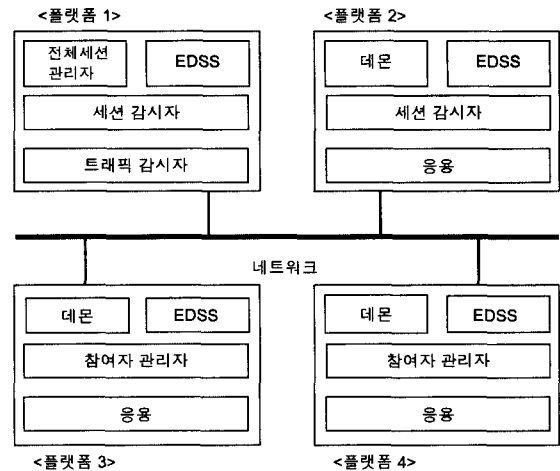
오류 공유가 가능하게 하기 위해서는 호 서버, 응용 및 오류 공유 서버, 웹 URL 공유 서버 등이 있어야 한다. 호 서버에서는 윈도우와 응용 사이의 사건 큐(event queue)에 사건 처리기와 사건 재지향기, 사건 여과기를 설치한다. 이

들은 사건 큐를 지나가는 사건들을 윈도우로부터 웹 응용 프로그램으로 전달되는 과정에서 가로채서 가공한 다음에 다시 윈도우나 응용 프로그램으로 전달한다. 이 과정에서 오류 공유 시스템이 필요한 처리를 행한다.

### 3.3 ESS\_WMCE의 분산 복제형 구조

#### 3.3.1 분산형 구조

두레 계층은 상호 참여형 멀티미디어 일반적인 응용을 개발하기 위해서 설계된 프레임워크이다. (그림 2)와 같다. 두레 계층은 EDSS, 전체 세션관리자(GSM), 세션관리자(SM), 서비스 에이전트들로 구성된다. 두레에서 제공되는 서비스 에이전트에는 여러 개의 기능들을 가진다. 플랫폼 1은 전체세션관리자(Global Session Manager)가 활성화되어 있으며 그 안에는 세션 감시자(Session Monitor)와 트래픽 감시자(Traffic Monitor)가 활성화되어 실행되고 있는 상태이다. 플랫폼 2는 세션의 초기 생성자인 세션관리자가 위치하는 곳이다. 플랫폼 2, 3, 4에는 데몬(Daemon)이 각각 하나씩 있는데 이것은 응용 사용자에게 세션관리 서비스를 제공하기 위한 가장 기본적인 세션관리 서비스 객체이다. 이 데몬 생성시 동시에 오류를 감지하여 자동으로 복구시켜주는 EDSS도 생성된다. 하나의 세션이 생성되면 이 세션에는 하나의 전체 세션 관리자, 하나의 세션 및 트래픽 감시자, 하나의 세션 감시자, 다수의 참여자 관리자가 생성된다. 여기서 전체세션관리자, 세션 관리자 및 데몬은 분산되어 있고 참여자 관리자와 응용은 복제되어 있는 분산 복제형 혼합 구조이다.

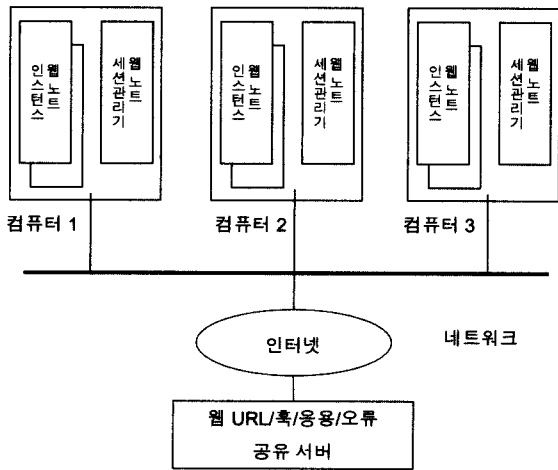


(그림 2) 근거리통신망에서의 단일 세션

#### 3.3.2 웹에서의 복제형 구조

각 사용자의 컴퓨터에 실행된 웹 노트 사이에 웹 페이지가 바뀔 때마다 브라우저 혹은 프레임 단위의 URL 주소 등의 네비게이션 정보를 전달해주어 서로 동일한 웹 페이지를 참조할 수 있도록 한다. 이러한 웹 노트는 복제형 구조에 기반한 응용 공유의 동작 원리를 이용하여 동기화를 수행하고 있다. 한 사용자가 동시에 여러 개의 세션에 참가

할 경우에 웹 노트 세션 관리기에 의해서 웹 노트 인스턴스가 복수 개로 생성이 되며 각 웹 노트 인스턴스는 세션 서비스 관리기를 가지고 있다. 웹 노트 세션 관리기는 새로 할당된 자원의 정보를 이용하여 세션 서비스 관리기를 생성하며 생성된 세션 서비스 관리기는 웹 노트 인스턴스를 구성하는 나머지 요소들을 생성 및 초기화 과정을 실시한다. 하나의 웹 노트 인스턴스는 다시 웹 노트 인스턴스 관리기에 의해서 여러 개의 웹 페이지 인스턴스가 발생된다. 하나의 웹 노트 인스턴스가 여러 개의 웹 페이지 인스턴스로 구성되는 경우는 우리가 인터넷 브라우저를 통하여 웹 문서를 볼 때 하나의 창이 새로 추가되면서 여러 개의 윈도우가 생성되는 경우를 볼 수 있는데 이 경우 웹 페이지는 각각 윈도우별로 동기화가 이루어져야 한다. 웹에서의 복제형 구조는 (그림 3)과 같다.



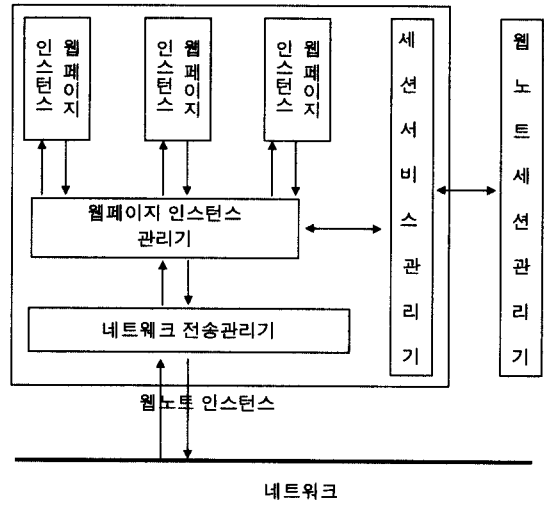
(그림 3) 웹에서의 복제형 구조

3.4 웹에서의 URL 및 오류 공유

3.4.1 웹 URL 및 오류 공유 서버

웹 URL 및 오류 공유 서버의 구성은 (그림 4)와 같다. URL 동기화 세션 관리기는 새로 할당된 자원의 정보를 이용하여 세션 서비스 관리기를 생성하며 생성된 세션 서비스 관리기는 URL 동기화 인스턴스를 구성하는 나머지 요소들을 생성 및 초기화 과정을 실시한다. 하나의 URL 동기화 인스턴스는 다시 URL 동기화 인스턴스 관리기에 의해서 여러 개의 웹 페이지 인스턴스가 발생된다. 하나의 URL 동기화 인스턴스가 여러 개의 웹 페이지 인스턴스로 구성이 되는 경우는 우리가 인터넷 브라우저를 통하여 웹 문서를 볼 때 하나의 창이 새로 추가되면서 여러 개의 윈도우가 생성되는 경우를 볼 수 있는데 이 경우 웹 페이지는 각각 윈도우별로 동기화가 이루어져야 한다. 새로운 윈도우의 생성이 필요할 경우 웹 페이지 인스턴스 관리기는 웹 페이지 인스턴스를 생성하고 이를 관리하기 위해 웹 페이지 인스턴스 식별자를 생성한다. 그 후 네트워크 전송 관리기를 통하여 웹 페이지 인스턴스가 새로 생성되었음을 알려주어 세션을 구성하는 다른 사용자의 웹 페이지 인스턴스 관리

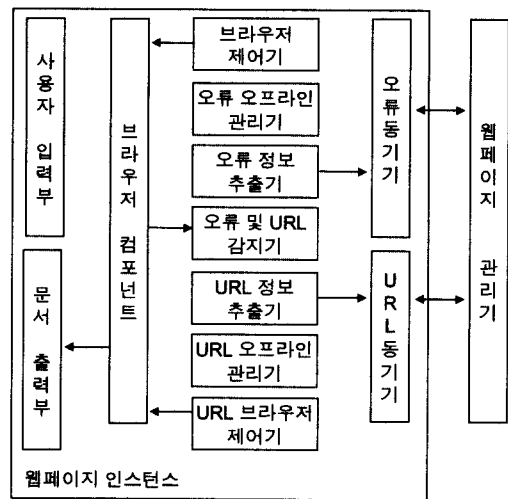
기로부터 웹 페이지가 추가되도록 하고 웹 페이지 인스턴스 관리기는 마찬가지로 웹 페이지 인스턴스 관리기를 생성한다.



(그림 4) 웹 URL 및 오류 공유 서버의 구성

3.4.2 EDSS의 구조

(그림 5)는 오류 공유의 블록 구성도이다.



(그림 5) 오류 공유의 기능 블록 구성도

본 논문에서 제안하는 웹 페이지 인스턴스는 크게 문서 출력부, 사용자 입력부, 브라우저 컴포넌트, 브라우저 제어기, 오류 및 URL 감지기(EUD)와 오류 및 URL 정보 추출기, 오류 및 URL 동기기(EUS) 등으로 구성된다. 즉, EDSS는 여러 기능의 에이전트가 존재하며 원활한 오류/URL 변경 감지 및 분류, 오류/URL 동기화 기법을 수행하는 멀티 에이전트 시스템이다. EDSS를 구성하는 구성 모듈로는 EUD (Error URL Detection)와 EUS(Error URL Sharing)이다. 여기서 브라우저 컴포넌트는 마이크로소프트사의 인터넷 익스플로러 등이 이용된다.

### 3.4.3 EUD

EUD는 오류 발생 및 URL 변경을 감지하는 핵심 에이전트로 윈도우의 훅킹(hooking) 방법을 이용하여 그 상태를 분석하여 오류 발생/URL의 변경 여부를 감지한다. 세션의 복원을 진행하기 위해서는 먼저 오류 및 URL 감지를 하기 위한 방법이 필요하다. 사용자는 사용자 입력부를 통해 오류 발생 및 해당 URL로 변경하고, 문서 출력부를 통하여 처리된 오류 또는 변경된 URL 페이지를 보게 된다. 오류 및 URL 감지기에서 훅킹 방법을 사용하여 사건의 발생을 감지한다. 오류가 발생하거나 사용자가 변경할 URL을 사용자 입력부를 통하여 URL을 변경하게 되면 브라우저 컴포넌트가 해당 URL로 브라우징을 실행한다. 이 때 감지기에서 받은 오류나 변경된 URL 정보를 오류 또는 URL 감지기가 변경된 사실을 인식하게 되어 오류 또는 URL 정보 추출기에 알려주게 된다. 오류 감지 방법에 대한 개요(scheme)는 다음과 같다.

Set of Detection = {Set of error, Set of error detector}

여기에서

Set of error = {E, D}

- E : 발생하는 오류
- D : 발생하는 오류의 도메인 위치

Set of error detector = {Addr\_EUD, Method\_EUD, Func\_EUD}

- Addr\_EUD : EUD의 주소 정보, 즉  $E_i(j)$ 에 대한 정보
- Method\_EUD : 윈도우에서 포인팅하는 함수를 가로채서 전달하는 방식인 훅킹 방식 사용
- Func\_EUD : EUD의 기능(function)은 두 집합 P,  $E_i(j)$ 에서  $R_1$ 을 집합 P에서  $E_i(j)$ 로의 관계(relation)라 한다.

$$R_1 = \{(p_i, e_i(j)) \mid (p_i, e_i(j)) \in R_1\}$$

관계  $R_1$ 에서는 키보드, 마우스, 시스템 등의 사건을 메시지 형태로 전송한다. 이 메시지를 훅킹 함수라고 한다. 사용자의 키보드나 마우스 같은 입력은 메시지 큐에 저장되고 윈도우는 명령어에 대한 훅 테이블을 유지하고, 실행할 함수를 포인팅 한다. 오류를 감지하는 방법도 이와 비슷한 과정을 거친다. 이 과정에서 오류를 감지한 내용, 즉, 포인팅하는 함수를 가로채서 전달하는 방식이다. EUD는 저널 레코드 또는 셸 훅 등의 훅킹 메시지를 사용하여 오류를 감지한다.

### 3.4.4 EUS

EUS는 오류 및 URL 정보 추출기는 오프라인 관리기를 통하여 정확한 오류 정보 또는 URL을 얻어오게 되는데, 만일 오프라인 모드로 동작 중일 때는 오프라인 관리기가 원격지 환경에 적용될 수 있는 올바른 오류 정보나 URL로 전환하여 오류 또는 URL 동기기를 통해서 원격지에 보내지게 된다.

Set of Error Synchronization = {Set of fault, Set of error synchronization}

여기에서

Set of fault = {F, S}

- F : 오류의 원인이 되는 고장(fault)
- S : 오류의 공유 여부

Set of error synchronization = {Addr\_EUS, Method\_EUS, Func\_EUS}

- Addr\_EUS : EUS의 주소정보,
- 즉  $E_i(j)$  및  $S_i(j)$ 에 대한 정보
- Method\_EUS : 오류 공유 방식 사용
- Func\_EUS : EUS의 기능(function)은 세 집합 P,  $E_i(j)$ ,  $S_i(j)$ 에서  $R_1$ 을 집합 P에서  $S_i(j)$ 로의 관계(relation)라 하고,  $R_2$ 를 집합  $E_i(j)$ 에서  $S_i(j)$ 로의 관계(relation)라 하면, 집합 P에서  $S_i(j)$ 로의 합성관계  $R_1 R_2$ 는 다음과 같이 정의된다.

$$R_1 R_2 = \{(p_i, s_i(j)) \mid (p_i \in P, s_i(j) \in S_i(j), (p_i, e_i(j)) \in R_1, (e_i(j), s_i(j)) \in R_2)\}$$

관계  $R_2$ 에서 웹 페이지 인스턴스 관리기로부터 오류 또는 URL 정보를 수신한 원격지 참여자의 URL 동기기는 브라우저 제어기에 보내고 오프라인으로 동작 중일 경우에 브라우저 제어기에 의해 자신에 맞는 URL로 변환된 URL 정보 또는 오류 정보는 브라우저 컴포넌트를 통해서 분석되어 문서 출력부에 보여지게 된다. 사용자가 웹 페이지 인스턴스의 URL 입력창 또는 URL 링크와 같은 사용자 입력부에 변경할 URL 또는 오류 정보를 입력하게 되면 해당 호스트가 제어가 가능한지 여부를 판별하여 제어가 불가능한 경우는 아무런 동작을 하지 않고, 제어가 가능한 경우는 브라우저 컴포넌트가 브라우징을 실시한다. 이 때 오류 감지기 또는 URL 감지기는 브라우저의 현재 오류 정보나 URL이 변경된 사실을 감지하고 오프라인의 경우는 오프라인 관리기를 이용하여 오류 정보나 URL을 원격지 환경에 적용 가능하도록 변환을 한다. 오류 동기기 또는 URL 동기기는 위의 과정에 의해 추출된 오류 정보나 변경할 URL 정보와 웹 페이지 인스턴스 관리기가 부여한 웹 페이지 인스턴스 식별자를 이용하여 패킷을 구성하고 웹 페이지 인스턴스 관리기를 통해 네트워크 전송 관리기로 전달하고 원격지로 패킷을 전송하게 된다.

오류 정보나 URL이 수신되면 해당 호스트가 제어 가능한지 여부를 판별하게 되는데, 이 때 제어가 가능한 경우는 자기 자신이 보낸 패킷이므로 아무런 동작을 하지 않는다. 하지만 제어 불가능한 경우는 오프라인으로 동작 중인지 여부에 따라서 오프라인인 경우 자기 자신에 적당한 오류 정보나 URL로 변환을 한 후에 브라우저 컴포넌트를 이용하여 오류 정보나 변경된 URL로 브라우징을 하게 된다. 동기화된 결과는 웹 브라우저 화면에 출력된다.

4. 시스템 평가

제안된 시스템은 Visual C++로 설계 및 구축하였다. 오류 감지에서 제안된 방법의 나은 점을 DEVS 형식론을 이용하여 모델링 및 시뮬레이션을 통하여 비교하였다. DEVS (Discrete Event System Specification)는 Bernard P. Zeigler에 의해 개발된 이산 사건 모델들의 계층 구조적 모듈화 방법을 제공해주는 형식론이다. 시스템을 작은 모듈들로 나누고 그것들로 전체 시스템을 계층적으로 구성해 나간다. 각 모듈들은 원자(atomic) 모델로 표현되며 그것들의 계층적 구성은 커플(coupled) 모델로 표현된다. DEVS 형식론에서 가장 기본이 되는 모델인 원자 모델은 다음과 같은 집합으로 표현된다[20].

- $M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, t_a \rangle$
- X : 외부 입력 사건들의 집합
- S : 상태 변수들의 집합
- Y : 외부 출력 사건들의 집합
- $\delta_{int}$  : 내부적 상태 변환 함수
- $\delta_{ext}$  : 외부적 상태 변환 함수
- $\lambda$  : 출력 함수
- $t_a$  : 시간 진행 함수

원자모델을 결합하여 새로운 커플모델을 형성한다.

본 논문에서는 전체 응용에 대해서 기존 방식 중의 하나인 폴링 방식을 사용하는 방식과 본 논문에서 제안한 혹킹 방법을 사용하여 오류 감지 시간을 줄이는 방식 2가지를 비교하여 효율성 검토를 한다. 시뮬레이션 모델을 통한 관측 목표와 관측 값 계산에 관련된 변수를 상태 변수로 가지는 모델이다. 오류 감지에 대한 효율성 비교는 <표 2>와 같다.

<표 2> 오류 감지에 대한 효율성 비교

대상 (object)	관 측 지 수		관측 모델
	기존 방식	제안된방식	
각 작업 (Each job)	ED <sub>1</sub> 은 EF에 RA <sub>1</sub> 반응시간 총합을 준다(종료 시점에서)	ED <sub>2</sub> 는 EF에 HA <sub>2</sub> 는 반응시간 총합을 준다(종료 시점에서)	transducer

[기존 방식]

DEVS 형식론에서 기존 방식의 변수를 정의하면 다음과 같다. 원자 모델은 EF, RA<sub>1</sub>, ED<sub>1</sub>이고 이 원자 모델들을 결합하여 새로운 커플모델을 형성한다. 처음에 외부 입력 사건들 중에서 폴링 시간을 받아들인다. 그 값이 RA<sub>1</sub>에서 입력 값이 되어 실행 한 후에 그 결과 값이 ED<sub>1</sub>의 입력 값이되어서 그 결과 값이 EF의 transducer를 통해서 관측될 수 있다. EF의 상태 변수는 poll-time<sub>1</sub>, RA<sub>1</sub>의 상태 변수는 pr-cnt<sub>1</sub>, ED<sub>1</sub>의 상태 변수는 tat-t<sub>1</sub>이다.

[제안된 방식]

DEVS 형식론에서 제안된 방식의 변수를 정의하면 다음

과 같다. 원자 모델은 EF, HA<sub>2</sub>, ED<sub>2</sub>이고 이 원자 모델들을 결합하여 새로운 커플모델을 형성한다. 처음에 외부 입력 사건들 중에서 혹킹 방법에 의한 감지 시간을 받아들인다. 그 값이 HA<sub>2</sub>에서 입력 값이 되어 실행 한 후에 그 결과 값이 ED<sub>2</sub>의 입력 값이되어서 그 결과 값이 EF의 transducer를 통해서 관측될 수 있다. EF의 상태 변수는 polltime<sub>2</sub>, HA<sub>2</sub>의 상태 변수는 pr-cnt<sub>2</sub>, ED<sub>2</sub>의 상태 변수는 tat-t<sub>2</sub>이다.

기존 방식과 제안된 방식의 비교는 다음과 같이 할 수 있다. 오류 감지 부분에서 기존 방법의 감지 시간(tat-t<sub>1</sub>)은 다음과 같다. 만일 프로세스 간의 메시지가 전달될 때 걸리는 시간을 t라고 하면 한번 폴링 시간은 2t가 된다.

$$tat-t_1 = 2t \times pr-cnt_1$$

제안된 방법의 감지 시간(tat-t<sub>2</sub>)은 다음과 같다.

혹킹 방법에 의해서 사건(event)이 발생하면 프로세스의 수에 관계 없이 1번 발생한다.

$$tat-t_2 = t \times 1$$

pr-cnt<sub>1</sub> > 1이므로 제안된 방법이 더 효율적이다. 프로세스의 수가 많아질수록 제안된 방식이 더 효율적이다. 만일 프로세스 간의 메시지가 전달될 때 걸리는 시간을 0.01초라고 하면 한번 폴링하는 시간은 0.02초가 된다. 응용 프로그램의 개수를 10, 20, 30, 40개라고 하면 기존 방법의 오류 감지 수행 시간은 각각 0.2, 0.4, 0.6, 0.8이 된다. 혹킹을 사용한 제안된 방법은 프로그램 개수에 관계없이 각각 0.01씩 된다. 프로세스의 수가 많아질수록 제안된 방식이 더 효율적임을 알 수 있다. 기능면에서 기존방법보다 제안된 방법의 나은 점을 <표 3>과 같이 비교하였다.

<표 3> CSCW에서의 오류 동기화 적용 비교

	Shastra	MERMAID	Mmconf	CECED	본 논문
구조	분산형	분산형	집중형 또는 복제형	복제형	분산 복제형
응용 공유	지원 안됨	지원 안됨	지원 안됨	단일 응용	지원
오류 감지	지원 안됨	지원 안됨	지원 안됨	지원 안됨	지원
오류 공유	지원 안됨	지원 안됨	지원 안됨	지원 안됨	지원
웹 노트	지원 안됨	지원 안됨	지원 안됨	지원 안됨	지원
통신 프로토콜	TCP/IP	TCP/IP	TCP/IP (multi-cast)	TCP/IP (multi-cast)	TCP/IP (multi-cast)

5. 결 론

본 논문에서는 URL 동기화 기능을 갖는 멀티미디어 공동 작업 환경에서 웹 세션을 잘 유지하기 위하여 오류 공유를 위한 시스템을 제안하였다. 사용자들 간의 공통된 웹

기반 문서를 참조할 수 있도록 웹 노트 서버의 개념을 도입시키고 이를 통해서 URL 및 오류 정보를 일치시켰다. 즉, 사용자들 간에 모두 같은 URL 및 오류 정보를 가지도록 한다. 따라서, 기존 프레임워크에서는 웹 노트 기능을 갖는 멀티미디어 공동작업 환경에서 웹 페이지와 미디어 등의 프로세스(process) 요소들에 대한 오류를 미리 감지하여 알려주고 오류 동기화 시스템을 이용하여 신속하고 정확하게 전달하여서 복구하는데 빠르게 대처할 수 있는 시스템이다. 오류 공유가 가능하게 하기 위해서 혹 서버, 응용 및 웹 URL, 오류 공유 서버 등을 제안하였다. 혹 서버에서는 윈도우와 응용 사이의 사건 큐(event queue)에 사건 처리기와 사건 재지향기, 사건 여과기를 설치한다. 이들은 사건 큐를 지나가는 사건들을 윈도우로부터 응용 프로그램으로 전달되는 과정에서 가로채서 가공한 다음에 다시 윈도우나 응용 프로그램으로 전달하였다. 이 과정에서 오류 공유 시스템이 필요한 처리를 하였다. 향후 연구 과제로는 웹 기반 멀티미디어 공동작업에서의 다중 세션, 네스티드 세션을 위한 오류 에이전트에 대한 연구 등이다.

**참 고 문 헌**

[1] 이재호, "협력 작업을 위한 에이전트 기반 소프트웨어", 한국 정보과학회 논문지, 제16권 제7호, pp.24-30, 1998.  
 [2] 이성제, 신근제, 김연준, 김문석, 성미영, "공동브라우저를 지원하는 웹 기반의 동기적 원격 학습 시스템", 한국 멀티미디어 학회 논문지, 제4권 제5호, pp.283-288, Oct., 2001.  
 [3] Makoto Kobayashi, Masahide Shinozaki, Takashi Sakiri, Maroun Touma, Shahrokh Daijavad, Catherine Wolf, "Collaborative Customer Services Using Synchronous Web Browsing Sharing," ACM conference on CSCW, pp.99-108, Nov., 1998.  
 [4] Dae J. Hwang, "DBM based Integrated Multimedia Distance Education System," In proceeding of International conference on On-Line EDUCA, Seoul, Korea, May, 1996.  
 [5] Dae J. Hwang, "Real Time Multimedia distanceeducation system," In proceeding of International conference on IEEE, October 1996, Beijing, China.  
 [6] 이경남, 김명균, 김진수, "웹 기반 RMON 에이전트 시스템 설계 및 구현", 2001년 한국 정보처리학회 춘계 학술발표 논문집, 제8권 제1호, pp.615-618, Apr., 2001.  
 [7] Kyung-Ha Kim & Dae-Joon Hwang, "The URL synchronization running on Distance Collaborative Video Conference with Lab Note Function," IEEE ISCE'98, Oct., 1998.  
 [8] Hiroaki Higaki, Kenji Shima, Takayuki Tachikawa, Makoto Takizawa, "Checkpoint and Rollback in Asynchronous Distributed Systems," IEEE INFOCOM'97, Proceedings Vol.3.

[9] 고응남, 황대준, "멀티미디어 공동작업 환경에 실행되는 오류 공유 시스템", 한국정보과학회2001 봄 학술발표논문집 (A), 제28권 제1호, pp.136-138, Apr., 2001.  
 [10] 장순주, 임종규, 정구영, 구용완, "분산 시스템에서 결함 허용성을 위한 프로세스 이주연구", 한국 정보 과학회지 가을 학술발표 논문집, Vol.21, No.2, p.132, 1994.  
 [11] 김문희, "결함 허용 시스템의 설계 고려사항 및 동향", 한국 정보과학회지, 제11권 제3호, p.7, 1993.  
 [12] 윤재영, 김학배, "Rollback과 Roll-forward기법을 사용한 TMR 고장의 시간여분 복구 정책", 한국정보처리학회논문지, 제6권 제1호, pp.216-224, Jan., 1999.  
 [13] Jonson, b.w. "Design and Analysis of Fault-Tolerant Digital Systems," Addison Wesley, 1989.  
 [14] Randell, B., "System Structure for Software fault tolerance," IEEE Trans, on Soft Engr., pp.220-232, June, 1975.  
 [15] A. Anupam and C.L. Bajai, "Collaborative Multimedia Scientific Design in Shastra," Proceeding of the ACM Multimedia'93, pp.447-456, Aug., 1993,  
 [16] T. Ohmori and K. Watabe, "Distributed Cooperative Control for Application Sharing Based on Multiparty and Multimedia Desktop Conferencing Systems : MERMAID," 4th IEEE ComSoc International Workshop on Multimedia Communications, pp.1-4. April, 1992.  
 [17] Torrence Crowley and Raymond Tomlinson, "MMConf : An Infrastructure for Building Shared Multimedia Applications," CSCW'90 Proceedings, Oct., 1990.  
 [18] Earl Craighill and Keith Skinner, "CECED : A System For Informal Multimedia Collaboration," Proceedings ACM Multimedia '93, pp.1-6. August, 1993.  
 [19] 박길철, "멀티미디어 상호 작용을 위한 객체 지향 협력 작업 프레임워크에 관한 연구", 성균관대학교 박사학위 청구 논문, p.12, 1997.  
 [20] Bernard P. Zeigler, "Object-Oriented Simulation with hierarchical, Modular Models," Academic Press, 1990.



**고 응 남**

e-mail : sskn@cheonan.ac.kr  
 1984년 연세대학교 수학과(이학사)  
 1991년 숭실대학교 정보과학 대학원 전산 공학과(공학석사)  
 2000년 성균관대학교 대학원 정보공학과 (공학박사)

1983년~1993년 대우통신 컴퓨터 개발부 선임 연구원  
 1993년~1997년 동우대학 전자계산과 교수  
 1997년~2001년 신성대학 컴퓨터 계열 교수  
 2001년~현재 천안대학교 정보통신학부 교수/멀티미디어 기술사  
 관심 분야 : 멀티미디어, 인터넷, 결함 허용, 에이전트 등