

□ 기술개설 □

분산 멀티미디어 서비스 개발을 위한 객체지향 비주얼 데이터 흐름도 작성 도구

한국전자통신연구소 임헌규* · 김두현* · 이재영** · 정찬근** · 임영환***

● 목 차 ●	
1. 서 론	3.3 데이터 흐름도의 일치성 유지
1.1 멀티미디어 응용 프로그래밍 동향	3.4 그룹공동 작업 영역(ShArea)
1.2 멀티미디어 비주얼 프로그래밍 동향	4. VDFC의 구조 및 구현
2. 분산 멀티미디어 데이터 처리 모델	4.1 사용자 인터페이스
2.1 클라이언트 서버 모델	4.2 내부 구성
2.2 가상객체인터페이스	4.3 동작 모델
2.3 스트림	5. VDFC의 사용 예
3. 객체지향형 비주얼 데이터 흐름도 작성도구	5.1 음성 및 영상 녹화
3.1 멀티미디어 데이터 처리 기본 모듈	5.2 원격 공동 영화 감상
3.2 데이터 흐름 경로	6. 결 론

1. 서 론

1.1 멀티미디어 응용 프로그래밍 동향

멀티미디어 데이터 입출력 관련 하드웨어의 상용화가 급진전되고, 통신속도가 빨라짐에 따라 분산 멀티미디어 응용 프로그램의 개발이 활발하게 진행되고 있다. 현재까지 상용화된 멀티미디어 데이터 처리 소프트웨어들의 대부분은 로컬 시스템 내에서의 저작도구 형태로 프레젠테이션 생성을 위한 도구들이 대부분이다. 그러나 최근에는 분산된 멀티미디어 데이터를 이용하는 프레젠테이션 생성 도구들의 개발이 활발히 진행되면서 분산 정보 접근 방법에 대한 많은 연구가 진행되고 있다[1-5]. 또 다른 응용 프로그램의 분야로서 영상회의를 기본으로하는 그룹웨어 관련 소프트웨어에 관한 연구가 활발히 진행되고 있다[6-8]. 이들

영상회의 시스템들이 그룹웨어를 위한 소프트웨어 플랫폼을 제공하여 그룹 공동작업시에 필요한 자료를 상호 교환할 수 있는 융통성 있는 환경을 제공하고 있으나, 일반적인 분산 멀티미디어 응용 서비스를 위한 플랫폼으로는 부족하다. 이와같은 멀티미디어 응용 소프트웨어들은 대부분이 라이브러리 형태로 제공되는 기능을 이용하여 직접 하드웨어에 대한 제어를 시행하며 데이터의 전송 및 동기화 등에 대한 루틴을 포함하고 있다[9].

라이브러리 함수를 이용하여 멀티미디어 응용 프로그램을 개발하는 경우는 프로그래머가 미디어 동기, 전송, 실시간 처리 등을 고려하여 설계를 해야 하며, 따라서 프로그램의 구성이 복잡하게 된다. 이러한 문제점을 해결하기 위해 멀티미디어 데이터 처리의 기본적인 기능을 제공하는 멀티미디어 데이터 처리 시스템 소프트웨어의 개발이 진행되고 있다[10-13]. 이와 같은 멀티미디어 데이터 처리 서버를 이용하여 프로그래밍 하는 경우에 프로그래머는 서버의

*정 회원
 **비 회원
 ***중신회원

기능을 액세스하기 위해 제공되는 API를 이용해야 한다. 이와같은 API를 이용하여 프로그래밍 하는 것도 쉽지는 않다. 이를 위해 비주얼 프로그래밍 도구에 대한 연구도 일부에서 진행되고 있다[14,15].

1.2 멀티미디어 비주얼 프로그래밍 동향

분산 멀티미디어 데이터 처리 프로그램의 개발 환경으로서 API를 사용하지 않고 시각화된 오브젝트들을 이용하여 시각적으로 프로그램을 구현할 수 있는 비주얼 프로그래밍 도구에 대한 연구는 비교적 활발하지는 않다. 앞에서 언급된 바와 같이 단지 분산 멀티미디어 프레젠테이션 작성을 위한 도구, 혹은 그룹 공동 작업을 위한 영상회의 프로그램의 환경 설정 등을 위한 비주얼 도구 등 특정 어플리케이션 영역에 국한된 형태의 도구에 대한 연구가 대부분이다.

비주얼 프로그래밍 언어 측면에서 볼때 멀티미디어 데이터 처리 응용프로그램을 위한 시각 언어는 처리 오브젝트를 시각화 하는 방법에 따라 분류할 때 데이터 흐름도 기반의 언어를 선택하는 것이 적절하다. 즉 데이터 흐름도 기반 시각 언어가 멀티미디어 데이터 처리와 같은 실시간 병렬 처리의 특성을 요하는 시스템을 위한 적절한 방법[16]이기 때문이다. 또한 멀티미디어 데이터 처리 모듈들 사이에는 정해진 프로토콜에 따라 데이터를 주고받는 독립적인 관계를 갖고 있으므로 프로그래밍 언어의 패러다임으로 분류할 때는 오브젝트 기반(Object Oriented) 프로그래밍 언어가 가장 적절하다고 볼 수 있다.

VDFC(Visual Data Flow Configurator)는 간단한 형태의 사용자 인터페이스를 제공하며, 객체 지향형 프로그래밍을 할 수 있다. 즉 VDFC에 의해 작성된 데이터 흐름도는 프로세싱 모듈에 해당하는 오브젝트와 이들 오브젝트를 연결하는 스트림 오브젝트로 구성되고, 이 흐름도에 대한 사용자 인터페이스에 의해 수행시에 제어가 가능하다. 프로세싱 모듈은 MuX(Multithread Cross Road)에서 제공하는 오브젝트에 대한 비주얼 아이콘으로 나타나며, 시스템의 하드웨어 및 관련 드라이버와 MuX를 위한 DLO(Dynamic Linking Object)가 첨부

됨에 따라 확장 가능하다. 이와 같은 VDFC의 기능은 멀티미디어 데이터 처리에 대한 비주얼 프로그래밍 도구로 기술이 가능하다. 또한 인터프리터가 제공됨으로써 구성된 멀티미디어 데이터 흐름도가 직접 수행될 수 있다. 즉 인터프리터는 VDFC에 의해 구성된 하나의 멀티미디어 데이터 흐름도와 관련된 MuX API를 호출함으로써, 수행이 가능하며 전체적인 제어와 개별 오브젝트에 대한 제어가 가능하다.

본 논문에서는 위와 같은 분산 멀티미디어 프로그래밍을 위한 멀티미디어 데이터 처리 서버(MuX)에 대한 기본적인 개념에 대해 언급을 하고, 이를 기반으로한 객체 지향형 비주얼 흐름도 작성 도구의 개념을 소개한다. 또한 현재 개발된 VDFC의 세부 구현에 대해 내부 구성 및 구현 상에서의 여러 사항들에 대해 기술하고, 실제로 본 도구를 이용하여 프로그래밍 하는 과정을 상세히 기술한다. 끝으로 본 도구의 설계 및 구현상의 문제점 및 향후 계획에 대해 언급하기로 한다.

2. 분산 멀티미디어 데이터 처리 모델

2.1 클라이언트 서버 모델

MuX 모델이 목표로 하는 계산 환경은 그림 1과 같이 멀티미디어 정보 통신망 상에서 분산되어 수행되는 멀티미디어 관련 응용 프로그램 또는 멀티미디어 서비스를 위하여 각 사용자의 가정이나 직장 또는 품안에 멀티미디어 정보 통신용 단말기가 존재하고 멀티미디어 정보 통신 서비스를 위한 서버급 컴퓨터가 망 상에서 존재하는 멀티미디어 분산 환경이다. MuX 모델은 이러한 분산환경에서 멀티미디어 회의,

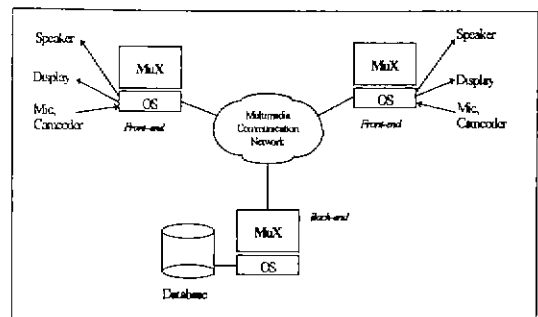


그림 1 MuX 모델이 목표표하는 분산 멀티미디어 환경

VOD(Video-On-Demand), 원격 교육, 원격 프리젠테이션, 하이퍼미디어 정보 검색, 전자 우편 등의 멀티미디어 통신 서비스를 위하여 사용자가 소유한 각 단말기나 서비스 서버 상에서 멀티미디어 입출력을 관리하고 멀티미디어 정보를 실시간에 처리하는 Front-End 및 Back-End 멀티미디어 서버를 위한 기본 계산 방식을 제시하는 것이다.

이러한 분산환경에서 멀티미디어 자원을 네트워크에 투명하게 사용할 수 있도록 하기 위한 방안으로는 MuX를 응용프로그램 라이브러리로 제공하는 방법과 서버 프로세스로 제공하는 방법이 있을 수 있겠다. 이 둘 중 현재의 MuX 시스템은 후자인 클라이언트-서버 컴퓨팅 방법을 따라 설계 구현되었다.

MuX가 클라이언트 서버 모델에 의하여 구현되었으므로 MuX를 사용하고저 하는 응용 프로그램 즉 클라이언트 프로세스는 먼저 통신망 상에 존재하는 MuX 서버와 접속하여야 한다. 이를 위하여 MuX의 크라이언트 라이브러리는 다음과 같은 객체지향형 API를 제공한다.

```
MuX :: MuX(char *host_name)
```

MuX 객체를 생성하므로써 서버와 접속되는 데 MuX 객체의 생성은 익히 아는 바 아래와 같이 실시한다.

```
MuX *hama_mux=new MuX("hama.etri.re.kr");
```

2.2 가상객체인터페이스

MuX의 모든 클라이언트 응용 프로그램은 C++ 클래스 중심으로 구성된 응용 프로그램 인터페이스 함수(API, Application Program Interface)를 이용하여 작성되어야 하는데 API는 가상 객체 인터페이스(Virtual Object Interface)를 사용하여 MuX 서버 내에서 멀티미디어를 처리하는 기본 모델을 거의 투명하게 반영하고 있다.

가상 객체 인터페이스는 서버와 클라이언트 간에 객체의 생성과 소멸, 그리고 소속함수의

호출 등에 관한 특수한 프로토콜을 통하여, 서버내에 실제적인 객체가 존재하도록 하고 클라이언트에게는 가상적인 객체를 만들어주어 클라이언트로 하여금 마치 실제 객체인양 사용할 수 있도록 하는 기법이다. 따라서 API로 설명되는 모든 객체는 가상객체라 할 수 있으며 실제 객체는 연결되어진 MuX 서버 내에 존재한다.

예를 들어 추후 설명할 원점 객체를 클라이언트가 사용하기 위해서 클라이언트 측에는 원점 가상 객체를 만들고 이에 상응하는 실제 원점객체는 서버에 만들어야 한다. 이를 위하여 MuX는 다음과 같은 형태의 API를 제공한다.

```
Source :: Source(MuX *mux_obj, char *DLO_name, unsigned long numeric_parm, char *string_parm)
```

그리고 상기의 API를 클라이언트에서 다음과 같이 사용할 수 있다.

```
Source *mic_source=new Source(hama_mux, "MicDlo", 0, NULL);
```

스트림은 단일 미디어의 흐름이다. 스트림이 처리하는 미디어의 종류는 다음과 같은 유형이 있을 수 있으며, 미디어 스트림은 이들 미디어에 대하여 화일, 디바이스, 네트워크, IPC, 분배기, 접합기, 분리기, 합성기 사이의 입출력 관계를 맺고 그 입출력 기능을 수행하여 미디어의 흐름이 형성되도록 한다.

2.3 스트림

스트림 층이 미디어의 흐름을 실현하기 위하여 세부적으로 다음과 같은 기능을 제공한다.

- 화일, 디바이스, 통신, IPC와의 입출력 서비스
- 미디어 분배기, 합성기, 분리기, 접합기와 의 입출력 서비스
- 스트림 필터링 서비스
- 다른 스트림과의 동기화를 위해 스트림으로 들어오는 데이터의 시표 표시

3. 객체지향형 비주얼 데이터 흐름도 작성 도구

멀티미디어 데이터 흐름도는 입출력 장치 및 내부 처리 모듈, 통신처리 모듈, 그룹 공동작업 영역 지원 모듈 등의 프로세싱 모듈과, 이들 사이의 데이터 흐름을 나타내는 데이터 흐름 경로(스트림)등 두 가지 오브젝트의 조합에 의해 구성된다. 이와 같은 개념을 그림 2에서 보여주고 있다.

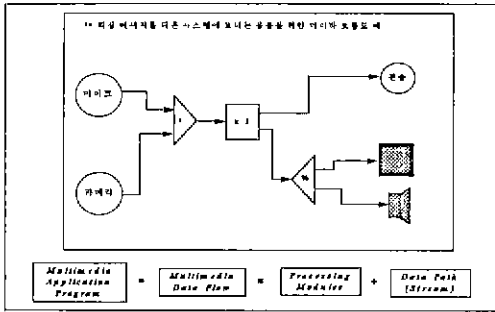


그림 2 멀티미디어 응용 프로그램과 흐름도

즉 하나의 멀티미디어 응용 프로그램은 멀티미디어 데이터 흐름도로 나타내어질 수 있으며, 흐름도는 멀티미디어 데이터 처리 모듈과 이들을 연결하는 데이터 흐름 경로인 스트림으로 구성된다.

3.1 멀티미디어 데이터 처리 기본 모듈

하나의 멀티미디어 데이터 흐름도를 구성하는 멀티미디어 데이터 프로세싱 모듈들은 응용 프로그램을 구성하는 프로세싱 모듈(함수)로서 인식될 수 있으며, 이들 프로세싱 모듈들 사이를 연결하는 데이터 흐름 경로는 함수들 사이의 데이터 전달 경로를 나타낸다. 또한 이들 프로세싱 모듈들은 각각이 독자적으로 수행되는 동적 오브젝트(Active Object)로서, 모든 프로세싱 모듈들은 하나의 스레드(Thread) 형태로 동작된다. 멀티미디어 데이터 처리 모듈들은 수행 특성에 따라 다음과 같은 종류로 구분될 수 있다. 하지만 이들 모듈들 사이의 연결시에는 이와 같은 구분은 특별한 의미가 존재하지는 않는다.

- 카메라, 마이크, 화일 등의 입력 장치 인

- 터페이스 모듈
- 출력 장치 인터페이스 모듈
- 실시간 데이터 전송 모듈
- 그룹 응용 프로그램을 위한 지원 모듈
- 데이터 스트림 변경 모듈
- 필터 모듈

입력 장치 인터페이스 모듈은 하나의 멀티미디어 응용 프로그램의 데이터 입력부로서, 연속적인 멀티미디어 데이터 스트림을 생성하는 데이터 소스(Data Source) 역할을 하는 모듈들이다. 출력 장치 인터페이스 모듈들은 일련의 처리 과정을 거친 최종 멀티미디어 데이터 스트림에 대한 출력을 담당하는 모듈로서, 비디오 데이터의 경우는 최종 영상 데이터를 화면에 출력하거나, 저장하는 등의 역할을 담당한다.

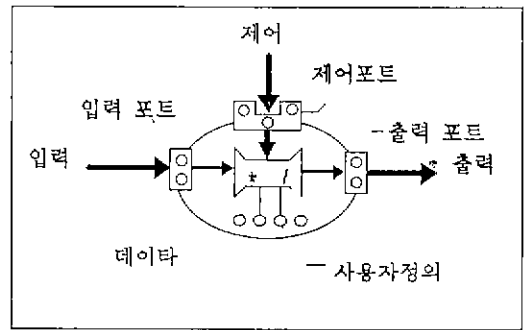


그림 3 멀티미디어 데이터 처리 모듈의 모델

하나의 멀티미디어 데이터 처리 모듈은 입력 포트를 통해 입력되는 스트림 데이터를 일련의 처리 과정을 거친 후에 출력 포트를 통해 실시간으로 연속적인 데이터 프레임을 출력하는 역할을 한다. 데이터의 처리는 내부 변수 및 변경 가능한 파라메타들에 의해 결정되며, 제어 포트를 통해 외부로부터의 제어 요청(사용자 동작)에 의해 제어가 가능하다. 각각의 프로세싱 모듈의 제어 기능은 서로 다르며, 이러한 기능에 대한 확인은 모든 데이터 처리 모듈이 공통적으로 포함하고 있는 제어함수 열람 기능에 의해 가능하다.

3.2 데이터 흐름 경로

멀티미디어 데이터 처리 모듈들 사이에는 연

속적인 멀티미디어 데이터 프레임이 전달되는 통로가 존재하는데 이를 데이터 흐름 경로라 하며, 이러한 데이터 흐름 경로는 VDFC에서는 하나의 연결선으로 표시된다. 이것은 하나의 회로도를 구성할 때 각 칩들 사이의 신호전달 회선을 말하며, 이들 연결선은 출력단과 입력단 사이의 데이터 타입 및 속도 등이 일치해야 한다.

하나의 입력 장치로부터 입력되는 멀티미디어 데이터는 최종 출력 모듈까지 여러 프로세싱 모듈을 거치며 변경될 수 있고, 따라서 중간간의 데이터 전송 경로의 포맷은 서로 다를 수 있다.

이들 데이터 스트림은 스트림 복제기(Stream Distributor), 분리기(Splitter), 합성기(Mixer) 등의 프로세싱 모듈에 의해 변경이 가능하다.

3.3 데이터 흐름도의 일치성 유지

완성된 하나의 데이터 흐름도가 수행되기 위해서는 일치성이 유지되어야 한다. 즉 서로 연결이 불가능한 데이터 전송 스트림이 없어야 한다. 구성된 흐름도를 수행되기 위해서는 다음과 같은 종류의 일치성 점검에 통과되어야 한다.

- 데이터 처리 모듈 파라메타의 일치성 점검
- 입출력 포트 수의 제한에 대한 점검
- 스트림의 미디어 종류 점검
- 스트림의 데이터 프레임 포맷
- Source 및 Destination에서의 데이터 프레임 처리 속도(Frame Rate)의 점검

이러한 일치성 유지는 주로 흐름도를 구성할 시점에 이루어진다. 흐름도를 구성하는 동안에 이루어지는 세부 일치성 점검은 다음과 같다.

위에서 언급된 일치성 검사 중 파라메타의 정확성 검사는 흐름도 수행시에 다시한번 이루어지며, 또한 서버의 API 호출시에도 서버에서 자체의 일치성 검사가 이루어진다. 한 처리 모듈의 파라메타값 변경이 일치성 유지와 관련된 경우는 변경 내용을 연결된 스트림을 따라가며 전파시키고, 변경이 요구되는 경우는 관련 처리 모듈들의 파라메타 값을 자동으로 변경해주는 기능이 포함되어 있다.

3.4 그룹공동 작업 영역(ShArea)

멀티미디어 데이터 흐름도는 그룹 회의와 같이 동적인 데이터 흐름의 변화가 요구되는 멀티미디어 데이터 처리 프로그램의 작성을 돕기 위해 Shared Area(ShArea)라는 프로세싱 모듈을 제공한다. 이것은 한 그룹의 공동 작업 환경에 필요한 기본적인 환경인 화상회의 뿐만 아니라 서로 공유하고 싶은 멀티미디어 데이터 스트림을 모든 멤버들에게 제공하기 위한 기본적인 환경으로써, 멀티미디어 데이터 스트림으로 표현되는 모든 데이터를 공유할 수 있다.

3.4.1 기본 개념

하나의 ShArea에 대한 개념은 일종의 회의실 탁자와 같은 것으로서, 사용자가 탁자에 둘러 앉아있는 모든 멤버들의 음성 및 모습을 듣거나 볼 수 있는 상태에서 한 멤버가 다른 멤버들에게 멀티미디어 데이터를 제공할 수 있는 환경을 기본으로 한다. 이와 같은 ShArea의 개념도를 그림으로 표시하면 그림 4와 같다.

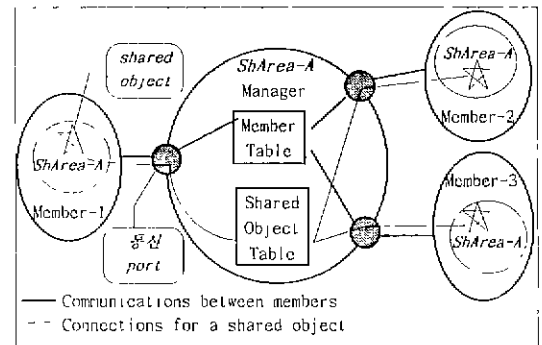


그림 4 ShArea의 개념도

ShArea Manager는 ShArea 서버로서 최초로 ShArea를 생성한 멤버가 사용하는 시스템에 존재하며 다음과 같은 기능을 한다.

- 참석 멤버들의 상태 관리
- ShArea 내의 shared object들에 대한 관리
- 상태 변화에 대한 알림
- 멤버들 사이의 메시지 전송

Member-1에 의해 제공되는 하나의 shared object는 ShArea에 다른 멤버들이 이용하기

위해 필요한 정보들을 알려주고, ShArea는 새로운 shared object에 대한 이용 가능성을 다른 멤버들에게 통보한다. 새로운 오브젝트에 대한 메시지를 받은 멤버는 상황에 따라 제공된 오브젝트로 부터의 데이터를 받거나, 무시할 수 있다.

ShArea에는 기본적으로 모든 멤버들의 음성 및 모습의 상호 연결을 위한 shared object 들을 포함하고 있다. 물론 이와 관련된 모든 오브젝트들은 흐름도를 구성하는 하나의 데이터 처리 모듈 오브젝트라 할 수 있다. ShArea 내에 존재하는 모든 처리 모듈의 출력은 해당 ShArea를 공유하는 그룹내의 모든 멤버들이 접근할 수 있고, 이를 제공하는 멤버는 이에 대한 사용 중지 및 변경이 가능하다.

하나의 shared object가 한 멤버에 의해 제공되면 이는 그림의 ShArea Manager에 의해 관리되며, 다른 멤버들의 접근은 ShArea Manager의 Shared Object Table로 부터 시작 된다. 이 테이블의 각 entry는 다음과 같은 종류의 정보를 포함하고 있다.

<i>Owner (제공자)</i>
<i>Object Name</i>
<i>Object-ID (제공자 시스템내의 Object ID)</i>
<i>Connection Style(1:N, N:N, N:1 중 하나)</i>
<i>Access Style(리소스 혹은 스트림)</i>
<i>Stream Type (Media 종류 및 프레임 포맷)</i>

그림 5 ShArea내 Shared Object의 정보

한 그룹의 멤버들이 서로 다른 흐름도를 구성한다 하더라도, 같은 ShArea를 공유하고 있으면 ShArea내의 모든 처리 모듈들은 대칭적으로 제공된다. 이들 ShArea 내에서 모든 멤버에 제공되는 공용 오브젝트(연속적인 멀티미디어 데이터 처리 모듈)들은 다음과 같은 특징에 의해 구분될 수 있다.

제공자와 사용자 사이의 관계

ShArea 내에 제공되는 오브젝트가 모든 멤버들 사이에 어떻게 제공되는가를 결정짓는 관계로서 다음과 같은 종류의 관계를 갖을 수 있다.

- 1 : N : Broadcasting
- N : N : Full Connection
- N : 1 : Collection

1 : N 관계는 한 멤버에 의해 제공되는 데이터가 모든 멤버에게 방송(broadcasting)되는 형태를 취한다. 이러한 관계를 갖는 공용 오브젝트는 모든 멤버들의 데이터에 대한 접근이 가능하나, 이에대한 제어는 제공자에게만 주어진다.

N : N 관계는 화상회의시에 기본적으로 필요한 모든 멤버들의 음성 및 모습을 멤버들이 동시에 듣거나 볼 수 있도록 하기위해 full connection 을 요구하는 관계를 말한다. 즉 영상의 경우에 모든 멤버는 자신의 영상을 다른 멤버에게 전달하기 위해 N : N 오브젝트에 연결해야 하며, 또한 다른 멤버들의 영상은 각 멤버를 나타내는 오브젝트에 의해 접근 가능하다.

끝으로 N : 1 관계는 모든 멤버들의 동일한 종류의 데이터를 수집하여 본인만이 이에 대한 제어 및 접근이 가능하도록 하는 오브젝트로서, 화상회의에서 음성의 연결이 이 관계에 해당한다.

3.4.2 ShArea 서버의 통신 프로토콜

ShArea를 이용한 그룹 응용 프로그램은 서버와 일련의 통신 프로토콜에 따라 다음과 같은 종류의 메시지 상호 교환을 위한 통신이 이루어진다.

- ShArea 참석 관련 메시지
- ShArea 상태 변화 관련 메시지
- ShArea 종료 관련 메시지

ShArea 서버와 그룹의 한 멤버가 사용하는 VDFC에 의해 구성된 흐름도 사이에는 다음과 같은 통신 프로토콜에 의해 그룹 공동 작업 영역에 대한 접근이 가능하다.

- ShArea 연결 요청(VDFC → ShArea 서버)
- ShArea 연결 허가(ShArea 서버 → VDFC)
- 멤버 등록 요청(VDFC → ShArea 서버)
- 멤버 정보 요청(VDFC → ShArea 서버)
- 멤버 정보 전달(ShArea 서버 → VDFC)
- 정보 변경 요청(VDFC → ShArea 서버)

- 멤버의 정보 변경 알림(ShArea 서버 → VDFC)
- 모든 멤버들에게 메시지 전달 요청(VDFC → ShArea 서버)
- 멤버의 메시지 전달(ShArea 서버 → VDFC)
- ShArea 사용 종료 허가 요청(VDFC → ShArea 서버)
- 멤버 종료 요청 Broadcasting(ShArea 서버 → VDFC)
- 멤버 종료 가능 전달(VDFC → ShArea 서버)
- 멤버 종료 허가(ShArea 서버 → VDFC)

위와 같은 통신 프로토콜에 의해 한 그룹의 모든 멤버들은 VDFC에 의해 구성된 흐름도에 의해 해당 그룹의 Shared Group Work Space에 접근할 수 있으며, 이를 통해 그룹 멤버들에게 오디오 및 비디오 데이터를 전달할 수 있다.

4. VDFC의 구조 및 구현

VDFC는 비주얼 프로그래밍 도구의 하나로써 멀티미디어 데이터 처리를 나타내는 오브젝트와 이들을 연결할 수 있는 사용자 인터페이스를 제공하고 있다. 본 절에서는 VDFC의 사용자 인터페이스의 설계 및 구현 상태와 내부 구성에 대해 언급하기로 한다.

4.1 사용자 인터페이스

본 도구의 사용자 인터페이스는 흐름도를 구성하는 기본 요소인 데이터 처리 모듈을 아이콘으로 표현하고 있으며, 현재 사용 가능한 멀티미디어 데이터 처리 모듈의 집합인 툴박스 윈도우가 별도로 존재한다. 사용자는 이 툴박스의 아이콘을 선택하여 흐름도 구성 윈도우에 생성하고, 두 오브젝트를 스트림으로 연결하는 작업과 각 오브젝트의 파라메타 설정만으로 멀티미디어 응용 프로그램인 흐름도를 생성할 수 있다.

VDFC의 사용자 인터페이스에 대한 예를 그림 6에 나타내었다.

사용자 인터페이스는 크게 4부분으로 나뉘

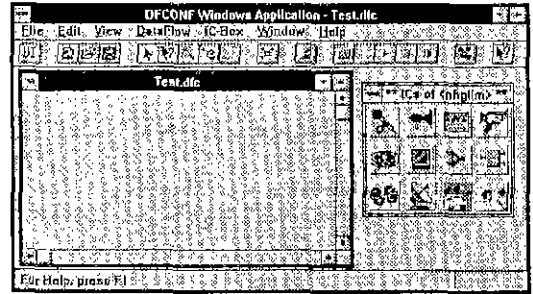


그림 6 VDFC의 사용자 인터페이스

다. VDFC를 위한 사용자 오퍼레이션을 선택할 수 있는 메뉴 및 툴바버튼을 포함하고 있는 주 윈도우와 하나의 흐름도를 나타내는 서브 윈도우 및 현재의 시스템에서 사용 가능한 멀티미디어 데이터 처리 모듈 및 VDFC에서 정의된 개념적인 처리 모듈들을 모아놓은 툴박스 윈도우가 존재한다. 메뉴 혹은 툴바 버튼을 이용하여 선택 모드로 지정한 후 툴박스 윈도우상의 임의의 아이콘을 선택한 뒤에 흐름도 구성 윈도우의 임의의 위치에서 마우스를 클릭함으로써, 멀티미디어 데이터 흐름도 상에 데이터 처리 모듈을 생성할 수 있다. 이렇게 생성된 흐름도 상의 데이터 처리 모듈은 파라메타 설정 모드로 지정된 후에 처리 모듈들의 내부 지정 가능 파라메타들의 값이 결정된다. 하나의 흐름도를 생성하는 처리 모듈의 생성 및 파라메타 셋팅이 완료된 후에 스트림 생성 모드에서 각 처리 모듈들 사이의 데이터 흐름을 지정할 수 있다. 이때 스트림의 연결에 대한 일치성 검사의 첫 단계가 수행된다. VDFC는 동시에 여러 개의 흐름도를 편집할 수 있으며, 각 흐름도 윈도우는 아이콘화 될 수도 있다.

현재 사용 가능한 처리 모듈을 모아놓은 툴박스 윈도우는 로컬 시스템 뿐만아니라 네트워크상의 다른 시스템에서 사용 가능한 처리 모듈을 나타내는 원격 시스템용 툴박스 윈도우를 여러개 나타낼 수 있다. 즉 주의의 여러 시스템의 멀티미디어 데이터 처리 모듈들에 대한 사용이 가능하며, 따라서 분산 처리가 가능하도록 되어 있다. 이러한 분산처리 기능은 MuX에서 기본적인 기능을 제공하고 있으며, 이를 시각화한 것이다.

4.2 내부 구성

VDFC의 내부 구성은 그림 7에 나타난 바와 같이 6개의 기능 블럭으로 구성되어 있다. 이들 기능 블럭들은 하나의 흐름도를 편집하는 것과 작성된 흐름도를 수행하기 위한 것으로 나뉜다. 데이터흐름도 인터프리터와 SAS (Shared Area Server) 인터페이스 블럭은 흐름도 수행을 위한 것이다. 수행과 관련된 모든 루틴들은 VDFC에서 제공되는 프로세싱 오브젝트 내에 존재하며, 새로운 오브젝트를 첨가하고자 하는 경우는 수행에 관련된 모든 루틴을 포함한 DLL을 필요로 한다.

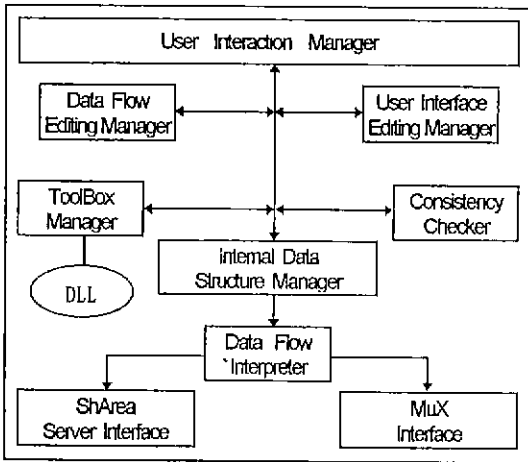


그림 7 VDFC의 기능블럭 구성도

User Interaction Manager는 흐름도의 구성 요소인 프로세싱 오브젝트와 이들 사이의 데이터 흐름을 나타내는 스트림을 가시화 하고 사용자의 동작에 대한 해석을 하는 모듈이다. Data Flow Editing Manager는 현재 구성중인 흐름도의 처리 모듈에 대한 관리를 하는 모듈이며, Consistency Checker는 새로이 생성되는 오브젝트 및 스트림의 일치성을 수시로 점검하는 모듈이다. User Interface Editing Manager는 구성된 데이터 흐름도에 대한 사용자 인터페이스를 편집하기 위한 모듈로서, 데이터 흐름도 상의 처리 모듈과의 관계설정 및 제어 함수들을 사용자 인터페이스 오브젝트와 연결하는 기능을 담당하는 모듈이다. Internal Data Structure Manager는 각 오브젝트에 대한 애트리뷰트를 관리하는 모듈로서,

Data Flow Editing Manager에 의해 호출되며 VDFC의 모든 편집(Editoring) 관련 처리 모듈 및 Data Flow Interpreter와 상호 보완 관계가 있다. 구성된 흐름도의 수행과 관련된 Data Flow Interpreter, MuX Interface 및 ShArea Interface는 수행중의 동적 제어(MuX API 호출) 및 ShArea 서버의 사용시에 서버와의 통신 및 통신 내용의 해석에 따른 처리를 각각 담당하는 모듈이다.

4.3 동작 모델

흐름도 기반 멀티미디어 응용 프로그램의 수행 모델은 분산 멀티미디어 데이터 처리 서버와 그룹 공동 작업 지원 서버, 응용 프로그램(Client)을 구성 요소로 하며, 이들 사이의 역할 분담 및 처리 관계는 그림 8에 나타난 바와 같다.

멀티미디어 응용 프로그램은 멀티미디어 데이터 처리 서버와 그룹 공동 영역 서버(SAS: Shared Area Server)등 두 시스템 소프트웨어의 지원을 받으며, 실제의 멀티미디어 데이터 처리 및 시스템의 멀티미디어 데이터 입출력 관련 자원(Resource)의 관리, 그룹 통신에 대한 기능의 수행을 두 서버에게 전담한다.

멀티미디어 응용 프로그램(Client)은 가상의 멀티미디어 데이터 처리 모듈(Virtual Object)만을 갖고 있으며, 주요 프로세싱은 사용자 인터페이스의 관리 및 유지와 사용자의 대화에

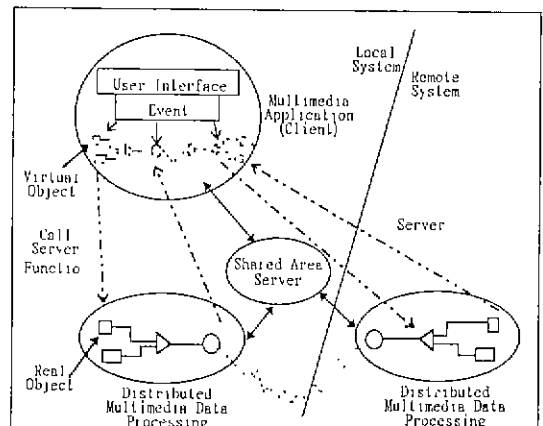


그림 8 VDFC에 의한 응용 프로그램 수행 모델

다른 Event 및 서버로부터의 Event에 대한 처리를 주로하고 있다. 즉 사용자의 동작에 의해 프로그램의 수행이 변경되는 경우는 Event Interpreter에서 이를 해석하고, 서버의 실제 오브젝트(Real Object)에 대한 제어는 서버에서 제공하는 API를 통하여 수행하게 된다.

그림에서 보여주는 바와 같이 그룹이 이용하는 멀티미디어 응용 프로그램의 경우에는 그룹 소속 멤버들 사이의 통신 및 멤버 관리등을 위한 Shared Area Server(SAS)가 필요하다.

5. VDFC의 사용 예

본 논문에서 제안된 객체지향형 비주얼 데이터 흐름도 작성도구의 프로토타입에 의해 구성 가능한 멀티미디어 데이터 처리 응용 프로그램의 구성 방법에 대해 화면을 중심으로 기술한다.

5.1 음성 및 영상 녹화

그림 9에 VDFC를 이용하여 자신의 음성 및 영상을 녹화하는 프로그램의 생성 예를 보여주고 있다.

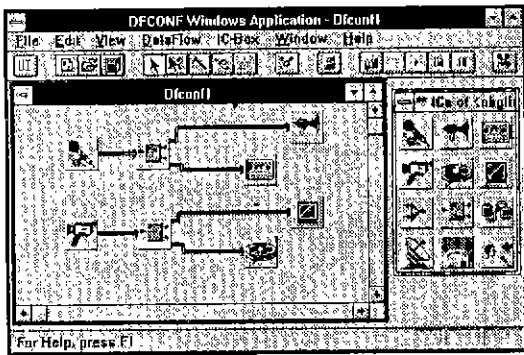


그림 9 음성 및 영상 녹화 프로그램의 데이터 흐름도 구성

마이크(카메라)로부터 입력되는 음성(영상) 데이터를 Duplicator를 이용해 두개의 스트림으로 복제하고, 하나는 자신의 스피커(윈도우)로 출력되게 함으로써, 녹화 상황을 인지할 수 있도록 하며, 동시에 하나의 출력은 화일에 저장하게 된다.

5.2 원격 공동 영화 감상

자신이 소유한 비디오를 다른 시스템의 사용자와 함께 감상하기 위한 멀티미디어 데이터 흐름도 구성 예가 그림 10에 나타나 있다.

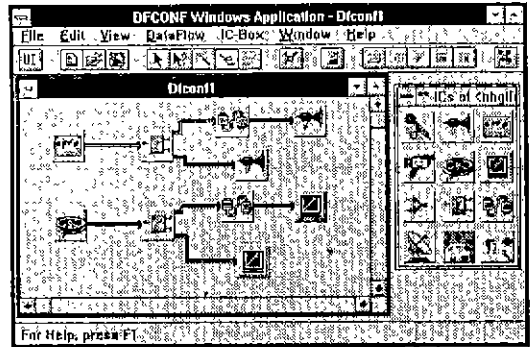


그림 10 원격 공동 영화 감상을 위한 데이터 흐름도 구성 예

6. 결 론

본 논문에서 제시된 멀티미디어 데이터 처리 응용 프로그램의 시각화 및 객체 지향형 비주얼 프로그래밍 도구의 일종이라 할 수 있는 데이터 흐름도 작성 도구는 프로그래밍 과정을 거치지 않고 시각화된 각종 멀티미디어 데이터 처리 모듈들을 데이터 흐름 개념에 따라 연결함으로써, 일반 사용자가 원하는 멀티미디어 데이터 처리 프로그램을 구현할 수 있는 기능을 제공하고 있다. 복잡한 형태의 응용 프로그램의 개발에 이용하기는 어렵지만 프로그래밍에 대한 기본 지식이 없는 일반 사용자가 자신의 컴퓨터 시스템의 하드웨어를 쉽게 활용할 수 있는 기능을 제공함으로써, 멀티미디어 컴퓨터의 활용 측면에서 볼 때 매우 유익한 도구이다.

VDFC의 기본적인 기능은 처리 모듈의 확장에 의해 그 기능이 계속 확장될 수 있으므로 새로운 하드웨어 및 소프트웨어의 확장이 비교적 쉽게 이루어 질수 있다. 또한 분산 처리를 위한 기본적인 기능을 제공하고 있으며, 여러 사용자가 동시에 수행할 수 있는 그룹 응용 프로그램의 개발을 위한 그룹 공동 작업 영역(ShArea)를 제공함으로써, 멀티미디어 데이터

의 공유 및 복잡한 멤버 사이의 관리에 대한 프로그래머의 작업을 덜어줄 수 있다.

현재까지 구현된 VDFC의 기능은 향후에 계속 확장될 예정이며, 아래와 같은 주요기능이 추가될 것이다.

- 사용자 인터페이스 편집기의 기능 보완 : 현재는 간단한 비디오 데이터 출력용 윈도우와 버튼등 몇가지 오브젝트만 제공되고 있으나, 복잡한 사용자 인터페이스의 구성을 위한 다양한 오브젝트를 제공할 예정이다.
- 그룹 공동 영역의 인터페이스 설정 기능 : 현재는 ShArea에서 제공되는 멀티미디어 데이터 스트림 및 공유되는 처리 모듈만이 제공되고 있으나, 공유되는 영역에 대한 사용자 인터페이스를 지정할 수 있는 기능을 부가함으로써, 공유 영역에 대한 일관된 접근이 가능하도록 할 계획이다.
- 데이터 흐름도의 처리 객체에 대한 그룹 설정 기능 : 하나의 데이터 흐름도를 개념적인 하나의 멀티미디어 데이터처리 객체로 인식할 수 있도록 흐름도의 일부 혹은 전체에 대한 그룹 설정 기능을 첨가할 계획이다. 이 기능이 첨가되는 경우 기존에 작성된 멀티미디어 데이터 흐름도를 하나의 처리 모듈로 보고 보다 복잡한 새로운 흐름도 작성에 이용할 수 있다. 또한 본 기능이 추가 되어야 프로그래밍 언어에서의 함수와 같은 개념이 제공됨으로 비주얼 프로그래밍 언어의 영역에 포함될 수 있을 것이다.

참고문헌

[1] B.Laurel, T.Oren, and A.Don, Issues in Multimedia Interface Design : Media Integration and Interface Agents, CHI'90, 1990, pp. 133-139.

[2] R.Cordes, H.Peyn, T.Kummerow, T. Topperwien, T.Weidenfeller. Access Methods for Distributed Multimedia Information Systems Based on Private Broadband Communication Networks. Multimedia92,

Monterey, CA, April, 1992, pp. 275-284.

[3] M.C. Buchanan, P.T. Zellweger, and Ken Pier, Multimedia Documents as User Interfaces, 1993, pp. 527-528.

[4] A.Sutcliffe and P. Faraday, Designing Presentation in Multimedia Interfaces, CHI '94, 1994, pp. 92-98.

[5] James Matthews, Peter Gloor, and Fillia Makedon. Videoscheme : A Programmable Video Editing System for Automation and Media Recognition. In Proceedings of ACM Multimedia 93, pp. 419-426, Anaheim, CA, August 1993.

[6] T.Ohmori, K.Maeno, S.Sakata, H.Fukuoka, and K.Watabe. Distributed Cooperative Control for Application Sharing Based on Multiparty and Multimedia Desktop Conferencing System : MERMAID. Multimedia92, Monterey, CA, April, 1992, pp. 112-131.

[7] M.S.Chen, T.Barzilai, and H.M.Vin. Software Architecture of DiCE : A Distributed Collaboration Environment. Multimedia92, Monterey, CA, April, 1992, pp. 172-181.

[8] S.Masaki, T.Arikawa, H.Ichihara, M. Tanbara, K.Shimamura. A Promising Groupware System for Broadband ISDN : PMTC. Multimedia92, Monterey, CA, April, 1992, pp. 190-198.

[9] Microsoft Corporation, Microsoft Video For Windows Users Guide. Microsoft Corporation, 1992.

[10] V.Phuah, J.Diaz-Gonzalez, R.Sasnett, S. Gutfreund. Developing Distributed Multimedia Applications. Multimedia92, Monterey, CA, April, 1992, pp. 199-206.

[11] R. Baker, A. Downing, K. Finn, E. Rennison, D.H. Kim, and Y.H. Lim, A Multimedia Data Processing Model for a Distributed Multimedia I/O System, Proceedings of the Third International Workshop on Networking and Operating System Support for Digital Audio and Video, San Diego, CA, November 1992.

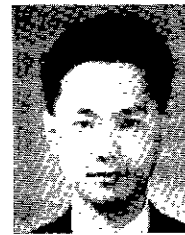
- [12] Brent Phillips. A Distributed Programming System for Media Applications. SM Thesis, Department of Electrical Engineering and Computer Science, MIT, July 1994.
- [13] D.L. Tennenhouse, J.F. Adam, D.Carver, H.Houh, M.Ismert, C.Lindblad. A Software-Oriented Approach to the Design of Media-Processing Environments. In Proceedings of the International Conference on Multimedia Systems and Computing, Boston, MA, May 1994. IEEE.
- [14] David J. Wetherall. An Interactive Programming System for Media Computation. MIT/LCS/TR-640, MIT, September 1994.
- [15] C. Lindblad, D.Wetherall, and D. Tennenhouse. The VuSystem: A Programming System for Visual Processing of Digital Video. In Proceedings of ACM Multimedia 94. ACM, October 1994.
- [16] S.K. Chang, Principles of Visual Programming Systems, Prentice-Hall, New Jersey, 1990, pp. 176.

김 두 현



1985 서울대학교 컴퓨터공학과 졸업(학사)
 1987 한국과학기술원 전산학과 졸업(석사)
 1993 정보커리기술사 취득
 1987~현재 한국전자통신연구소 선임연구원, 한국과학기술원 전산학과 박사 과정
 관심분야: 분산 멀티미디어 처리, 멀티미디어 시스템 성능 평가

이 재 영



1988 서울대학교 물리학과 졸업(학사)
 1990 The Johns Hopkins University 물리학과 졸업(석사)
 1994 The Johns Hopkins University 물리학과 졸업(박사)
 1994~현재 한국전자통신연구소 Post. Doc.
 관심분야: 그룹웨어, 시각인어, 분산 멀티미디어 처리

정 찬 근



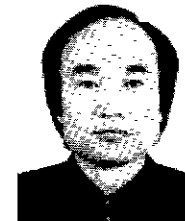
1979 한국항공대학교 전자공학과 졸업(학사)
 1981 서울대학교 전자공학과 졸업(석사)
 1983~현재 한국전자통신연구소 책임연구원
 관심분야: 분산 멀티미디어 처리

임 헌 규



1985 숭실대학교 전산학과 졸업(학사)
 1987 한국과학기술원 전산학과 졸업(석사)
 1987~현재 한국전자통신연구소 선임연구원, 한국과학기술원 전산학과 박사 과정
 관심분야: 그룹웨어, 시각인어, 분산 멀티미디어 처리

임 영 환



1977 경북대학교 수학과 졸업(학사)
 1979 한국 과학원 전산학과 졸업(석사)
 1985 Northwestern University 졸업(박사)
 1985~현재 한국전자통신연구소 책임연구원
 관심분야: 분산멀티미디어 처리