

## X 응용 프로그램의 개발을 지원하는 RAD 도구의 구현

최보철 · 박성휴 · 박태원 · 이명재  
컴퓨터 · 정보통신공학부

### <요 약>

X 윈도우 시스템과 위젯 집합들은 응용 시스템의 사용자 인터페이스 구성과 이벤트 처리를 위해 많은 자원을 제공하지만, 이들에 대한 코딩 작업은 많은 시간을 필요로 한다. 본 논문에서는 위젯 계층에 대한 기술문으로부터 계층구조를 자동으로 분석하고 Motif 위젯 집합 기반의 ANSI C코드를 생성하는 Motif 사용자 인터페이스 생성기와 폼 기반의 설계 작업을 지원하는 Rad 도구의 구현에 대해 기술한다. RAD 도구는 X 라이브러리를 사용하여 개발되었다.

---

## Implementation of an RAD tool supporting development of X Application Systems

Bo Chul Choi · Seong Hue Park · Tae Won Park · Myeong Jae Yi  
School of Computer Engineering and Information Technology

### <Abstract>

Although X window systems and many widget sets support many resources for constructing user interface and processing events in application systems, coding works for these are very time-consuming. In this paper, we describe our works for implementing a Motif User Interface builder which automatically analyzes hierarchy structures and produces Motif based ANSI C source code from the description of widget hierarchy, and an RAD tool which supports form based design. We have developed the RAD tool using X library only.

# 1. 서론

X 윈도우 시스템의 가장 중요한 특징 중의 하나는 장치 독립적인 구조(device-independent architecture)를 갖는다는 것이다. X는 문자나 그래픽을 포함하는 응용 프로그램들이 X 프로토콜을 지원하는 어떠한 하드웨어에서도 수정하거나 재컴파일 또는 재링크시키지 않고도 윈도우들을 디스플레이할 수 있도록 해준다. X 윈도우 시스템은 네트워크 투명성 윈도우 시스템으로서, X 응용 프로그램들이 네트워크로 연결된 어느 컴퓨터에서든지 수행될 수 있도록 해주며, X 프로토콜을 통해 클라이언트와 서버간에 통신을 하게된다. X의 이런 구조에 비추어 볼 때, X 프로토콜은 프로그래머의 관점에서 보면 X 시스템에서 제공되는 X 라이브러리로부터 생성된다고 볼 수 있다. X 라이브러리를 사용하수 있는 언어는 C, ADA, LISP, C++ 등의 언어가 일반적이고 C와 C++언어가 가장 많이 사용된다.

X 윈도우 시스템의 구조를 계층적으로 표현해 보면 그림 1과 같은 형태이다. X의 주요 구성요소인 X 프로토콜, X 라이브러리, X 툴킷(Toolkit) 등을 볼 수 있다.[1][2]

이런 X 라이브러리만을 가지고 GUI를 작성한다는 것은 많은 양의 코딩을 요구하며, 매우 복잡한 작업을 필요로 한다. X 윈도우 시스템에서는 사용자가 이러한 객체들을 체계적으로 편리하게 사용할 수 있도록 X 툴킷을 제공한다.

만일 버튼, 메뉴, 스크롤바 등과 같은 객체들이 미리 작성되어 있다면 사용자는 GUI작성 시 가져다 사용하면 된다. 이렇게 미리 작성된 객체들의 집합을 툴킷이라 한다. 이러한 툴킷은 여러 단체에 의해 여러 종류가 개발되었고 X 윈도우 시스템은 이러한 툴킷 중 하나와 연관되어 동작하게 된다.[3][4]

대부분의 툴킷은 두 부분으로 구성된다.

- X Toolkit Intrinsic (Xt Intrinsic라 부름)
- 사용자 인터페이스 구성요소, 일반적으로 위젯이라 한다.

현재는 여러 종류의 위젯들이 제공되고 있는데 MIT의 Athena 위젯, Open Software Foundation의 OSF/Motif위젯, AT&T에서 제공한 OPENLOOK 위젯 등이 있다.[5]

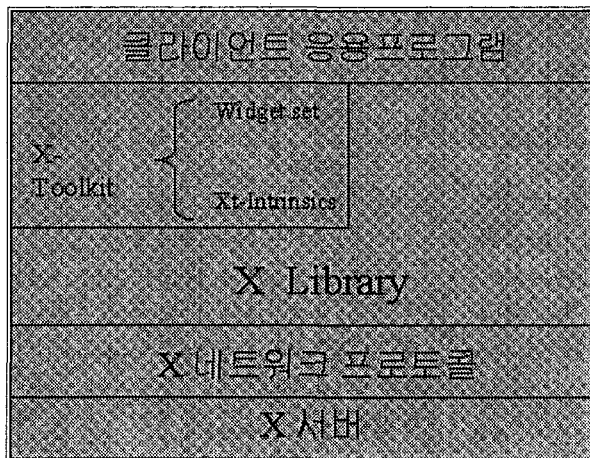


그림 1 X의 계층 구조

X 윈도우 시스템에서 Xt 툴킷을 기반으로 어플리케이션을 개발하는 것이 X 라이브러리만을 사용해서 개발하는 것보다는 비용과 효율성 면에서 월등히 낫다고 볼 수 있지만, 결국 원시 코드 레벨에서 작업하는 동안은 많은 양의 코드 편집을 필요로 한다는 한계를 벗어날 수 없다. 이러한 단순 노동적인 작업과 그에 따라 발생할 수도 있는 오류를 줄이기 위해서는 사용자 인터페이스 관리 시스템 혹은 시각적 개발 도구(Visual Development Tool)와 같은 도구가 필요할 수밖에 없다.

X 윈도우 상에서 X 툴킷을 사용하여 폼 기반의 설계(Form Based Design)를 지원하는 개발 도구를 구현한다는 것은 X 툴킷의 구조상 쉽지 않다. 왜냐하면, 대부분의 X 어플리케이션들이 윈도우 관리자(Window Manager)와 함께 작동하게 되므로 윈도우 관리자를 배제하고 어플리케이션을 사용할 수가 없기 때문이다.

X 기반 개발 도구에서는 응용 프로그램을 개발할 때에 사용자 인터페이스에 대한 설계 작업은 위젯들에 대한 계층 구조를 고려하게 된다. 그리고 그러한 계층 구조로 설계된 위젯의 속성과 자원을 적절히 수정하고 그 정보를 이용해서 원시 코드를 생성한다.

본 논문에서는 상업화되어 가장 많이 사용되며 Xt Intrinsics를 기반으로 하는 OSF/Motif 위젯을 이용하여, Motif 어플리케이션의 개발을 지원하는 Motif UI 생성기와, Motif를 사용하지 않는 일반적인 X 어플리케이션의 개발을 지원해주는 폼 기반의 X RAD 도구의 개발에 대해 논한다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 사용자 인터페이스 도구에 대한 기본 개념과 본 연구와 관련된 기존의 도구들에 대해 살펴보고, 3장에서는 Motif 응용 프로그램의 사용자 인터페이스 개발을 지원하는 Motif UI 생성기에 대해 기술한다. 이 도구는 Motif 위젯 집합을 사용하여 개발한 것으로 위젯 트리 기반의 설계를 지원한다. 4장에서는 폼 기반의 설계를 지원하여 주는 X RAD의 개발에 대해 기술한다. 이 도구는 X library만을 사용하여 개발하였다. 마지막으로 5장에서는 결론을 맺고 향후 연구 과제에 대해 기술한다.

## 2. 사용자 인터페이스 도구

### 2.1 사용자 인터페이스 도구

컴퓨터 프로그램의 사용자 인터페이스는 그 프로그램을 사용하는 사용자로부터 어떤 입력을 받아들이거나 그 사람에게 보여줄 출력을 제어하는 부분이다. 1992년 Myers의 조사에 의하면 평균적으로 응용 프로그램 코드 중에서 48% 정도가 사용자 인터페이스에 관련된 부분이며, 구현 시간의 거의 50% 정도가 사용자 인터페이스를 구현하는데 소요되었다고 한다.[6] 오늘날은 거의 모든 응용 프로그램들이 그래픽 사용자 인터페이스를 가지고 있다. 1993년의 조사 결과는 Unix 상에서 개발된 소프트웨어의 97%가 그래픽 사용자 인터페이스를 가지고 있다. 이와 같은 연구 결과에 비추어볼 때 그래픽 사용자 인터페이스를 개발하는 것은 응용 프로그램의 개발에 있어서 필수적이면서도 매우 많은 시간이 소요되는 부분이라는 것을 알 수 있다.

사용자 인터페이스 도구는 시간이 지나가면서 많은 이름으로 불리어졌는데, 그 중 가장 널리 사용된 이름은 사용자 인터페이스 관리 시스템(User Interface Management System,

UIMS)이다. 그러나 많은 사람들이 UIMS라는 용어는 일련의 동작들(사용자로부터 각각의 이벤트가 발생한 후에 일어나는 동작들)을 제어할 수 있는 도구에만 사용되어야 한다는 생각을 가지고 있기 때문에, 인터페이스 생성기, 인터페이스 개발 도구, 그리고 최근에는 RAD(Rapid Application Development) 도구라는 용어로 사용된다.

사용자 인터페이스 도구를 사용함으로써 얻을 수 있는 이점은 다음과 같이 두 가지 그룹으로 분류할 수 있다.[7][8]

- 인터페이스의 품질이 높아진다.

설계 작업이 빠르게 시제품화 되며, 구현될 수 있고, 사용자들의 시험 작업을 통해서 발견되는 변경 요구사항들을 쉽게 수용할 수 있다. 또한 동일한 어플리케이션이 여러 인터페이스를 갖는 경우나 서로 다른 어플리케이션이 비슷한 인터페이스를 갖는 경우가 많은데 기존의 작업들을 재사용할 수도 있고, 도구의 사용으로 인해 일관된 사용자 인터페이스를 갖게 된다.

- 사용자 인터페이스 코드를 만들고 유지보수하는 것이 쉬우며, 경제적이다.

인터페이스에 대한 명세서가 더욱 쉽게 표현되며, 검증되고, 평가될 수 있다. 또한 도구에 의해 코드의 많은 부분이 자동적으로 생성되므로 적은 양의 코드만 작성하면 된다. 사용자 인터페이스 부분과 어플리케이션의 다른 부분들과의 모듈화 작업이 지원되므로 유지보수가 쉬어진다.

## 2.2 X 인터페이스 생성기

본 연구에서 개발한 Motif 인터페이스 생성기와 X RAD 도구와 관련된 기존의 도구들은 다음과 같은 것들이 있다.[9][10]

X 윈도우 시스템이 널리 사용되면서, 대표적으로 상용화된 인터페이스 생성기 도구로는 TeleUSE와 UIMX를 들 수 있다. TeleUSE는 X 윈도우와 OSF/Motif를 사용하여 구축되었으며, 정적인 스크린 배치와 설계, 콜백에 대한 자동 구현, 실행 가능한 코드의 생성, 대화식 시험, 디버깅, 그리고 유지보수 작업을 포함하여 그래픽 사용자 인터페이스 개발에 대한 모든 단계들에 대해 지원을 해준다. UIMX는 Motif 코드와 함께 사용자 인터페이스 언어(User Interface Language, UIL)를 생성한다. 이것은 어플리케이션의 사용자 인터페이스 부분에 대한 코드를 대화식으로 생성, 수정, 시험할 수 있도록 해준다. 또한 여러 플랫폼들 사이(Motif, 윈도우스 등)에 쉽게 이식 가능한 인터페이스를 생성할 수 있고 기존의 키보드 지향적인 응용 프로그램에 대해 아이콘화된 인터페이스들을 생성할 수도 있다.[11]

X-Designer는 Motif와 MS Windows 모두를 위한 GUI 생성기이다. 하나의 설계물로부터 X/Motif나 MFC 라이브러리와 함께 생성될 수 있는 C 또는 C++ 코드를 생성해준다. WINTERP는 Xlisp 기반의 Motif 툴킷으로서 시제품의 빠른 생성과 대화식 프로그래밍을 지원한다. 이 도구는 OSF/Motif 위젯과 X 툴킷에 대한 고수준의 객체 지향 인터페이스이며, Xtango의 path-transition 애니메이션 패러다임에 근거한 그래픽/애니메이션 위젯, 그리고 다른 Unix 프로세스들과 데이터에 대한 통신을 위한 기능 등을 갖고 있다.

VXP(Visual X windows Programming interface)는 Yong Chen에 의해 개발된 것으로

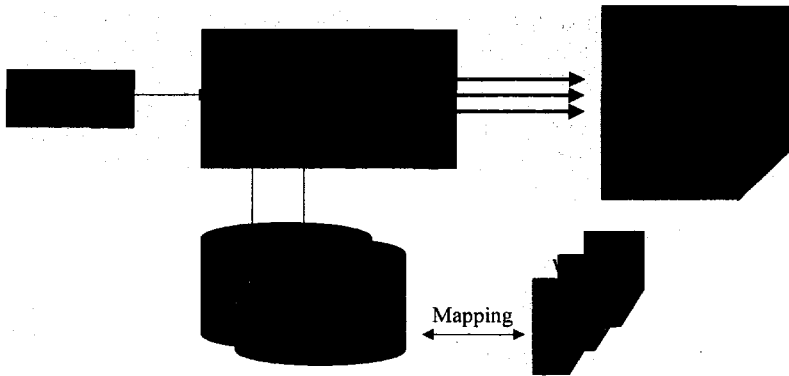


그림 2 원시 코드 생성기의 구조

서, 프리웨어이다. 이 도구는 Motif GUI 생성기로서 약간의 UIMS 기능도 갖고 있으며, SGI irix, HP hp-ux, Sun OS4와 Solaris 2.x 등에 포팅되었다. 그밖에 Common Lisp에 기초한 GUI 툴킷인 Garnet을 포함해서 많은 상업용 도구와 프리웨어들이 있다.

### 3. Motif 사용자 인터페이스 생성기

#### 3.1 구조

Motif 사용자 인터페이스 생성기는 Motif 어플리케이션의 사용자 인터페이스를 자동으로 생성하여 주는 도구로서, OSF/Motif 위젯 집합을 사용하여 C 언어로 구현하였다. 사용자는 자신이 생성할 어플리케이션의 사용자 인터페이스에 사용될 Motif 위젯들에 대한 계층 관계를 트리로 구성하여 이 도구에 입력으로 주게 된다. 생성기는 그런 트리에서부터 위젯의 관계를 추출하고 그것을 ANSI C Code로 변환하게 된다.( 그림 2 )

ANSI C코드로 변환하는 과정에서 ANSI C 코드로의 변환을 쉽게 하기 위해 중간과정으로 생성된 계층 트리에 사용된 위젯의 정보를 이용하여 간단한 스크립트 언어를 생성하게 된다. 이 정보는 트리를 쉽게 구성할 수 있도록 위젯 맵핑 테이블을 통해서 생성되는 위젯간에 연관성을 지어준다. 이것은 IR(Intermediate Representation)과 유사한 역할을 하게 된다. 이렇게 한 이유는 목적언어로 C 언어뿐만 아니라 다른 언어로도 쉽게 변환할 수 있도록 하기 위해서이다.

일단 설계 정보(위젯에 대한 계층 트리)가 생성되면 코드 생성기에 정보가 넘어가서 실제로 트리를 생성할 수 있는지를 검사하게 된다. 만약 타당한 정보일 경우 코드 패턴에서 실제 사용하게 될 목적 언어로 맵핑을 하게 되고, 맵핑된 자료는 대응되는 위젯 생성 코드를 가져와서 목적 언어의 규칙에 따라 원시 코드를 생성하게 된다. 원시 코드는 그림 2에 표시된 것과 같이 Motif 위젯 생성 함수인 `XmCreateWidget(...)`, `Xt Intrinsics`의 콜백 등록함수인 `XtAddCallback(...)` 과 같은 내용을 포함하게 된다.

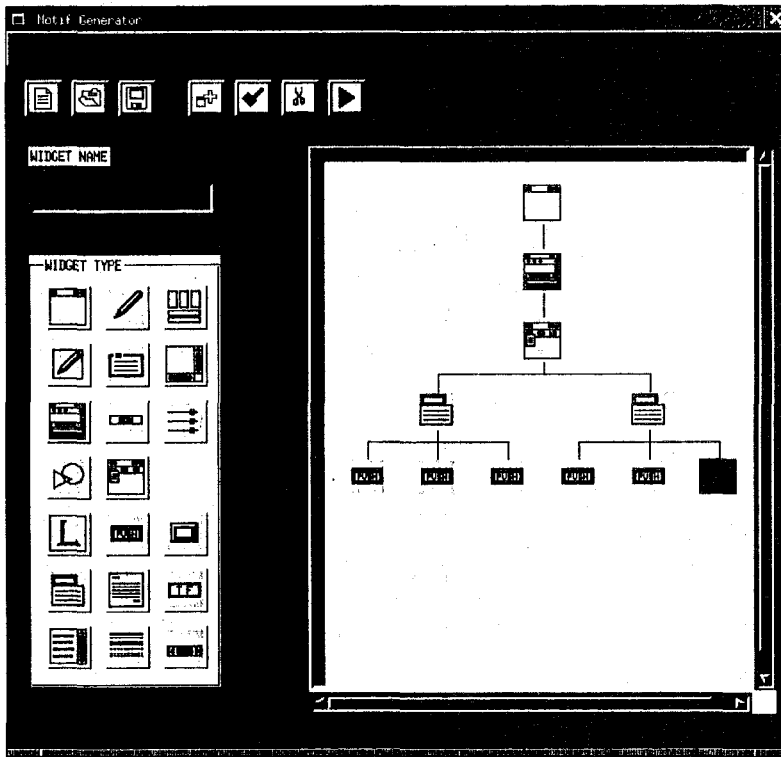


그림 3 사용자 인터페이스의 구성

### 3.2 사용자 인터페이스의 구성

일반적인 개발 도구일 경우에는 폼 기반의 설계 작업을 지원할 수 있지만, 본 도구는 Motif를 이용해서 개발하였고 Motif의 사용자 인터페이스 개발 작업을 지원하기 위해, 개발하고자 하는 사용자 인터페이스에서 사용될 위젯들에 대한 계층 트리 개념을 설계 도구로 사용하였다. 따라서 본 도구의 사용자 인터페이스는 (그림 3)에 표시한 것과 같이 크게 세부분으로 구성되어 있다.

왼쪽 아래 부분에 있는 것은 Motif 위젯 집합에서 제공하는 위젯들을 각각의 위젯들의 특성에 따라 아이콘으로 표시한 위젯 아이콘 테이블이다. 사용자는 여기서 자신이 개발할 사용자 인터페이스에서 사용할 위젯들을 선택하고, 선택된 위젯이 위젯 계층상의 어떤 부분에 위치할 것인지를 오른쪽 아래 부분에 있는 위젯 계층 트리에 삽입하게 된다. 즉, 사용자는 왼쪽에서 필요한 위젯을 클릭하여, 오른쪽에서 필요한 계층관계를 만들어 나가게 되어있어 한 눈에 위젯의 전체 구조를 살펴볼 수가 있다. 그리고 생성된 정보를 처리하는 메뉴바로 구성된다. 메뉴바에서 제공하는 기능은 기본적인 복사, 오리기, 붙이기 기능과 생성한 위젯 계층 트리로부터 목적 언어를 생성하게 하는 Generate 명령 등이 있다.

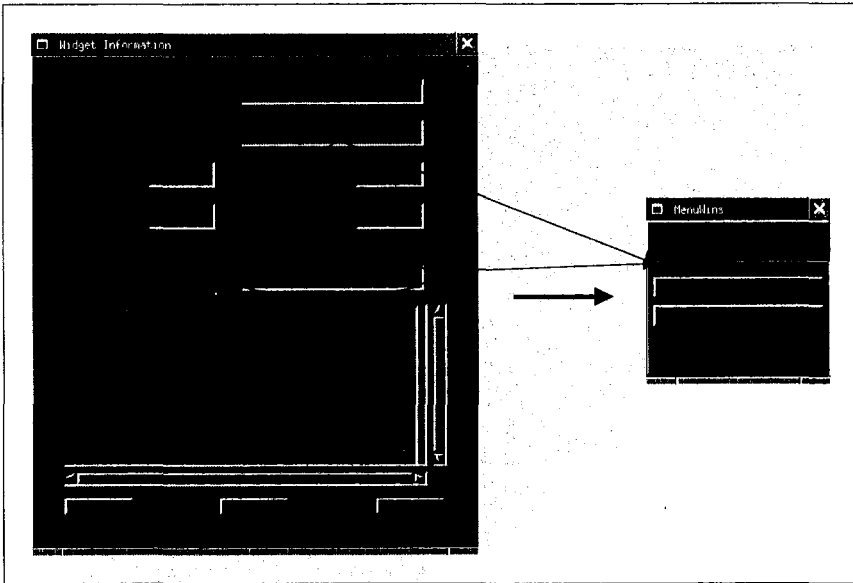


그림 4 자원 설정 과정

### 3.3 자원의 관리

X 서버의 자원은 윈도우, 비트맵, 폰트, 컬러 등을 포함한다. X 서버 레벨에서 자원은 하나 이상의 클라이언트를 대신해서 서버 내에서 관리되는 자료 구조들을 의미한다. 그런데 위젯의 경우 자원 관리가 꽤 까다로운 편이다. 여러 가지 방법으로 자원을 관리할 수 있기 때문이다. 자원 관리는 클라이언트 프로그램 내부에서의 자원 관리가 있을 수 있고 /usr/lib/X11/app-defaults 디렉토리에 의한 관리, <HOME>/.Xdefaults 파일에 자원을 지정하는 경우, 실행 명령어에서 선택사항으로 지정 등 일반적으로 4가지 중 하나로 관리하게 된다. 현재 구현한 Motif 사용자 인터페이스 생성기의 경우는 클라이언트 프로그램 내부에서 직접 자원 관리를 하도록 코드를 생성해준다. 그림 4는 그림 3과 같은 과정에 의해 생성된 응용 프로그램의 자원을 설정하는 과정을 보여주고 있다. 일단 응용 프로그램을 생성하는 과정에서 자원의 지정을 하게되면 코드가 생성되면서 자원의 정보를 그대로 이용하게 되므로 자원을 바꿔야 할 경우 손쉽게 수정을 할 수 있다.

### 3.4 어플리케이션의 사용자 인터페이스 개발 과정

사용자가 어플리케이션의 사용자 인터페이스를 개발하기 위해서는, 사용자 인터페이스를 구성하게 될 위젯들의 계층 관계를 고려하여, 위젯 아이콘 테이블에서 필요한 위젯을 선택하여 위젯 계층 트리에서 자신이 원하는 위젯들 간의 연관 관계(상위 위젯과 하위 위젯을 구분함)를 설정하면 된다. 또한 각각의 위젯에 필요한 자원을 설정한다. 그러면 위젯 트리에 관한 정보는 자연스럽게 위젯 정보와 맵핑되어 스크립트 언어로 저장되게 된다. 그런 다음 Generate 명령을 실행하게 되면, 저장되어 있는 스크립트 언어는 번역과 동시에 목적

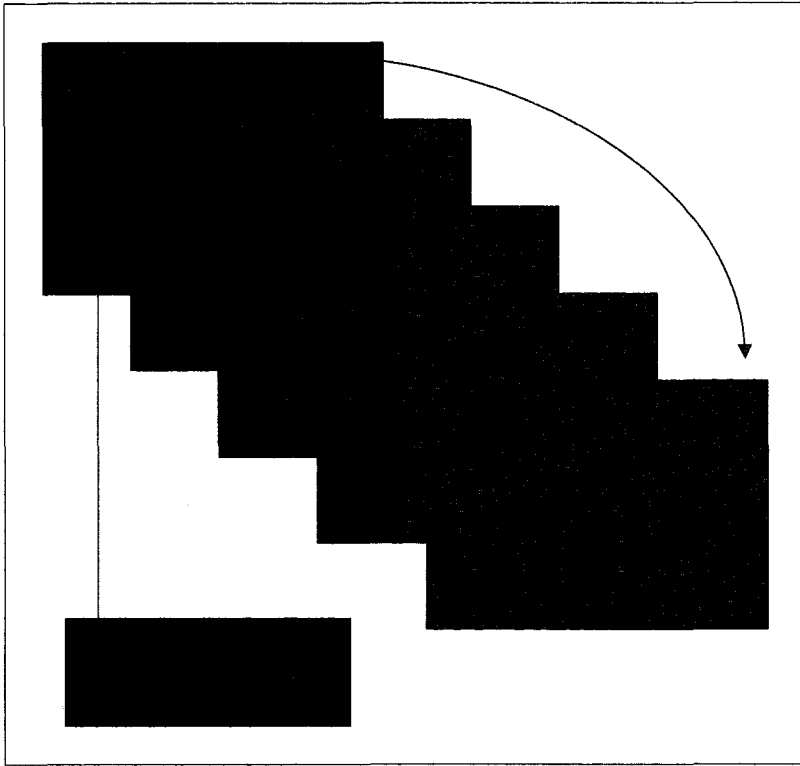


그림 5 사용자 인터페이스 개발 과정

언어로 생성되고, 동시에 컴파일까지 하게 된다. 그리고 컴파일이 끝나면 생성된 오브젝트 파일이 실행된다.(그림 5)

### 3.5 재사용성(Reusability)

이렇게 생성된 계층관계에 대한 정보만으로 새로운 어플리케이션을 손쉽게 생성할 수 있다. 이미 제작했던 응용 프로그램의 계층 관계만을 읽어 온 다음 정보의 내용을 수정하고 다시 재 컴파일 하면 커다란 노력 없이 새로운 응용 프로그램을 이전의 작업을 재사용하여 쉽게 개발할 수 있다. 이 작업은 대부분의 Unix 어플리케이션 특히 Motif 스타일의 어플리케이션들은 비슷한 사용자 인터페이스를 갖고 있기 때문에, 이들에 대한 사용자 인터페이스 개발시 거의 똑같은 내용을 중복해서 코딩하는 것보다는, 기존에 개발했던 어플리케이션의 사용자 인터페이스를 재사용함으로써 개발에 드는 노력을 절감할 수 있게 된다.



## 4. X-RAD 도구의 설계 및 구현

### 4.1 X-RAD (Rapid Application Development) 도구 개발 동기 및 개요

3장에서 기술했던 Motif 라이브러리를 이용해서 개발한 Motif 사용자 인터페이스 생성기는 Motif 어플리케이션의 사용자 인터페이스 개발에 많은 지원을 하고는 있지만, 크게 두 가지 면에서 보완할 부분이 있다.

첫째로는 Motif 사용자 인터페이스 생성기가 위젯 계층 트리 형태의 입력을 받아 사용자 인터페이스를 생성하고 있기 때문에 사용자가 Motif 위젯 집합에 대한 정확한 이해가 없이는 사용자가 원하는 형태의 그래픽 사용자 인터페이스를 만들어 내기 힘들다는 것이고, 두 번째로는 Motif 윈도우 관리자가 필요 없는 X 어플리케이션을 구축하고자 할 경우에는 X 라이브러리만을 가지고 작업을 해야하는데, 이 경우에는 여전히 처음부터 X 라이브러리를 가지고 수작업으로 사용자 인터페이스를 구축해야 한다는 것이다. 물론 두 가지 모두 Motif 사용자 인터페이스 생성기의 원래 기능이 Motif 어플리케이션의 사용자 인터페이스의 개발 지원이라는 원래 기능에 비추어보면 별 문제가 아닐 수도 있지만, 어쨌든 보다 쉽게 X 어플리케이션의 개발을 지원하기 위해 순수한 X 라이브러리만을 이용한 X-RAD 도구의 개발 필요성을 느꼈다. 그래서 순수한 X 라이브러리만을 이용하여, 트리 기반의 설계가 아닌 폼 기반의 설계를 지원하는 X-RAD 도구를 구현하였다. X-RAD 도구를 이용하면 사용자는 자신이 원하는 사용자 인터페이스를 직접 폼 기반으로 설계할 수 있고, 그러한 형식의 사용자 인터페이스를 생성하여 주는 원시 코드를 결과물로 얻을 수 있다.

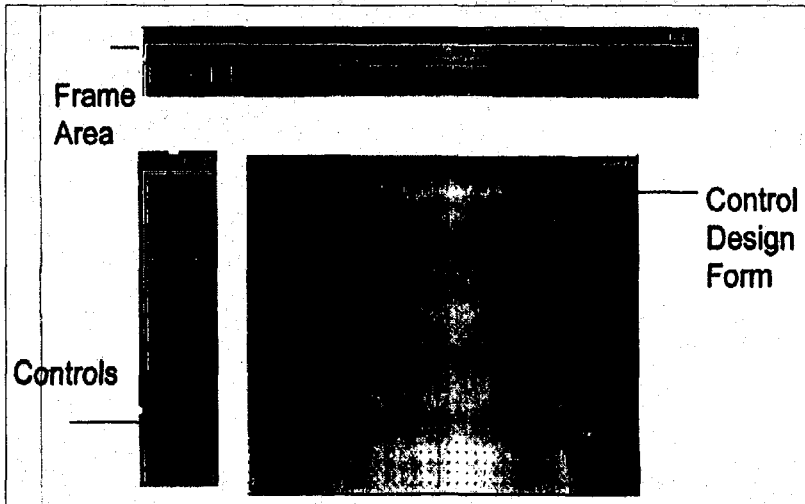


그림 6 X-RAD 도구의 사용자 인터페이스

X-RAD 도구의 사용자 인터페이스는 폼 기반의 설계를 지원하는 다른 개발 도구들과 유사하게 구성되어 있다.(그림 6) 이 도구는 폼 기반의 설계를 지원하고 있어서, 사용자는 왼쪽의 컨트롤 영역에서 자신이 원하는 컨트롤들을 선택한 후에 오른쪽의 컨트롤 설계 영역에서 자신이 원하는 위치에 배치하고 필요한 속성만 정의하면 된다. 이와 같이 사용자는 이미 개발되어있는 컨트롤들을 사용해서 쉽게 X 어플리케이션을 개발할 수가 있게 된다.

## 4.2 설계 및 구현

X-RAD는 오직 X 라이브러리만을 이용하여 개발해야 하고, 대부분의 어플리케이션에서 공통적으로 사용되는 사용자 인터페이스를 구성할 수 있도록 하기 위해, 먼저 사용자 인터페이스의 구성 요소들에 대한 분류 작업과 그것들의 개발을 지원하는 모듈들에 대한 구축 작업을 수행하였다.

현재 상용화되어 있고, 많은 사람들이 사용하고 있는 제품 중에서 MS의 MFC(Microsoft Foundation Class)를 보면 각각의 분업화되어있는 클래스들을 쉽게 발견할 수 있다. 이런 클래스들은 거대한 응용 프로그램을 작성할 경우 필수적인 프레임워크라고 볼 수 있다. 본 논문에서도 이런 응용 프레임워크를 제작하였고 그 프레임워크를 사용하여 X-RAD를 개발했다.

X-RAD 도구는 각각의 필요한 클래스들을 클래스 계층으로 분류하였고, X Intrinsic을 사용하지 않고 직접 X 라이브러리로만 구성된 클래스로 모든 구현을 하였다. 사용한 언어는 ANSI C++이며, 사용한 컴파일러는 g++이다.

## 4.3 클래스들의 구성

사용자 인터페이스의 개발을 지원하기 위해 필요한 클래스들을 구축하였는데, 구축한 클래스들을 어플리케이션 개발에 필요한 기능별로 분류해 보면 다음과 같이 크게 네 가지로 분류할 수 있다.

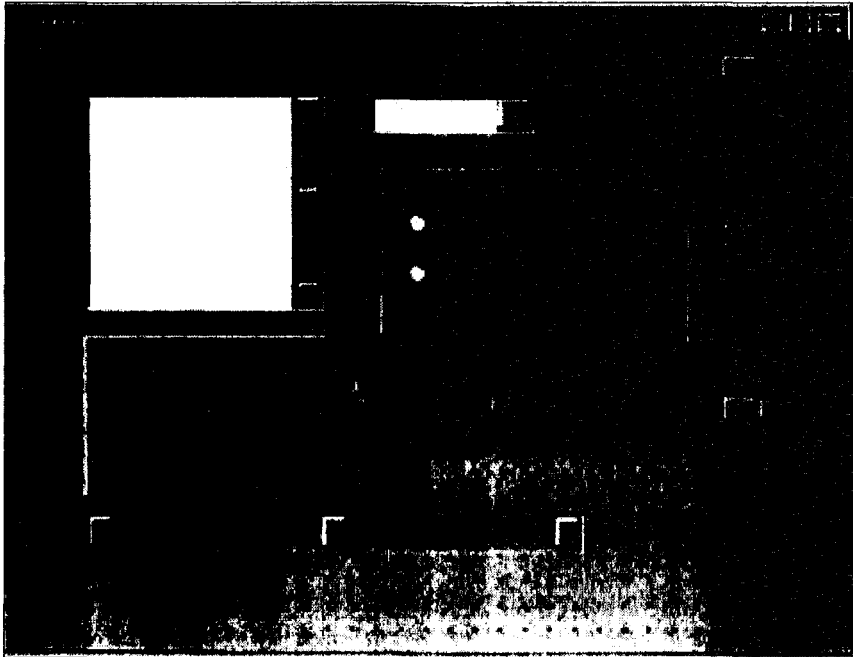


그림 7 X-RAD 도구를 사용한 간단한 예제

- 모든 클래스의 Top 클래스 - XObject
- X 어플리케이션의 프레임 구조 - XFrameWinApp  
XFrameWinApp 클래스는 윈도우의 위치, 색깔, 경계에 대한 정보를 포함해서 발생하는 Event를 처리하는 부분을 포함하고 있다.
- 독립 윈도우기능 - XDialog  
XDialog 클래스는 다이얼로그의 생성과 삭제에 관련된 속성과 메소드를 정의하고 있다.
- 사용할 컨트롤 - XControl  
XControl 클래스는 사용자에게 제공해줄 모든 컨트롤들에 공통적인 속성과 메소드를 정의하고 있다.

```

<SampleViewer.h>
...
#include "XWindowApp.h"
#include "XListBox.h"
#include "XButton.h"
#include "XHScrollBar.h"
#include "XVScrollBar.h"
#include "XLabel.h"
...
class SampleViewer : public XWindowApp {
...
private :
    XListBox*   m_listBox1;
    XButton*    m_button1;
    XHScrollBar* m_hScrollBar1;
    XVScrollBar* m_vScrollBar1;
    XLabel*     m_label1;
...
};
<SampleViewer.C>
#include "SampleViewer.h"
SampleViewer :: SampleViewer(INT _x, INT _y, UINT _width, UINT
_height) :
XWindowApp(INT _x, INT _y, UINT _width, UINT _height)
{
...
}
SampleViewer :: ~SampleViewer()
{
...
}
...

```

그림 8 X-RAD에 의해 생성된 SampleViewer 코드

그밖에 클래스들, 예를 들면 XForm, XIconTool 클래스들은 XDialog 클래스에서, 버튼, 체크 상자, 선택 상자, 리스트 상자, 레이블, 텍스트, 스크롤바, 툴바 등을 구현한 클래스들은 XControl 클래스에서 파생된다.

#### 4.4 X-RAD를 이용한 사용 예제

X-RAD 도구를 사용하여 그림 7과 같은 간단한 예제 SampleViewer를 만들어보았다. 컨트롤 영역에서 리스트 박스, 버튼, 스크롤바, 레이블들을 선택한 후에 그것들을 폼 설계 영역에 배치한 모양을 보여주고 있다. 이러한 폼 설계로부터 X-RAD 도구는 그림 8과 같이 header 부분과 body 부분의 두 개의 C++ 파일을 생성하게 된다.

### 5. 결론 및 향후 연구 과제

본 논문에서는 X상에서 Motif를 이용하여 개발한 원시 코드 생성기와 순수한 X 라이브러리만을 사용하여 개발한 RAD 도구의 구현에 대해 기술하였다. 원시 코드 생성기는 위젯 계층으로부터 C 원시 코드를 생성하여주며, RAD 도구는 폼 기반의 설계를 가능하게 하여준다.

UNIX환경은 일반 PC에서 사용하는 Windows계열의 GUI에 비해 열악한 점이 많다 특히 개발 Tool일 경우 많은 제약이 따른다. 설계에서부터 코딩, 그리고 컴파일까지 일일이 손이 가야하는 곳이 대부분이다. 하지만 본 논문에서 개발한 Motif 사용자 인터페이스 생성기나 X-RAD와 같은 개발 도구를 이용한다면 Unix 환경에서의 응용 프로그램 개발에서 사용자 인터페이스에 관련된 코딩 작업이 엄청나게 줄어들 것이고, 또한 비슷한 사용자 인터페이스들을 쉽게 재사용할 수 있다는 측면에서 이 두 개의 도구는 많은 기여를 할 것으로 기대된다. 향후 연구 과제로는 이 도구들을 실제 Motif 어플리케이션이나 X 어플리케이션의 개발에 적용할 수 있도록 신뢰도와 안정성을 높이는 작업과, 다른 환경의 어플리케이션 예를 들면 MS Windows 환경에도 맞는 어플리케이션의 개발 지원 그리고 Java와 같은 언어의 코드 생성 등 도구의 기능을 확장하는 작업이 진행되어야 할 것으로 생각된다.

#### 참고문헌

- [1] Douglas A. Young, The X Window System™ Programming and Application with Xt, 2nd Edition, PTR Prentice Hall, 1994.
- [2] Oliver Jones, Introduction to the X Window System, Prentice Hall, 1989.
- [3] Nye A. and O'Reilly T. X Toolkit Intrinsic Programming Manual, Motif Edition. O'Reilly and Associates, 1990.
- [4] David Flanagan. X Toolkit Intrinsic Reference Manual, 3rd Edition, O'Reilly and Associates, 1992.
- [5] Young D. Object-Oriented Programming with C++ and OSF/Motif, Prentice-Hall,

1992.

- [6] Brad A. Myers and Mary Beth Rosson. "Survey on User Interface Programming," In Human Factors in Computing Systems, Proceedings SIGCHI'92, Monterey, CA, May, 1992. pp. 195-202.
- [7] Brad A. Myers. "User Interface Software Tools," ACM Transactions on Computer-Human Interaction. vol. 2, no. 1, March, 1995. pp. 64-103.
- [8] Brad A. Myers, "State of the Art in User Interface Software Tools," Advances in Human-Computer Interaction, V.4, 1993. pp.110-150.
- [9] X Frequently Asked Questions, <ftp://ftp.x.org:/contrib/faqs/FAQ>
- [10] MOTIF Frequently Asked Questions: Subject 92-96, is available in <http://www.premier.sco.com/tech/MotifFAQ.html>
- [11] Visual Edge Software Ltd. UIM/X. 3950 Cote Vertu, Montreal, Quebec H4R 1V4. Phone (514) 332-6430. 1990.