

## 협업을 통한 시각 프로그램 개발 환경

조용윤<sup>0</sup>, 유재우  
송실대학교 컴퓨터학과

yycho<sup>0</sup>@ss.ssu.ac.kr, cwwoo@comp.ssu.ac.kr

### A Development Environment for Visual Program With Cooperation

Yong-Yoon Cho, Chae-Woo Yoo  
School of Computing, SoongSil University

#### 요 약

시각 프로그램은 프로그램의 내부 로직에 대한 코드와 외부 사용자와의 상호 작용을 위한 사용자 인터페이스에 대한 코드로 나뉘어 구성되어 있다. 개발자들은 빠르고 효과적인 시각 프로그램 개발을 위해 GUI(graphic user interface) 개발 도구를 이용한다. 그러나, 기존의 사용자 인터페이스 개발 도구는 여러 개발자들간의 능력과 개발 분야에 따라 시각 프로그램의 내부 로직과 인터페이스 부분을 분리해 공동 개발할 수 있는 기회를 제공하지 못한다. 본 논문은 다수의 시각 프로그램 개발자간 자신의 개발 분야에 따라 분산 공동 개발이 가능한 네트워크 기반의 시각 프로그램 공동 개발 시스템을 제안한다. 제안하는 공동 개발 환경은 개발자간 독립성과 협동성을 지원하고 즉각적이고 동기적인 개발 협의와 의견 교환을 위한 통신 창을 제공해 응용 프로그램의 개발 속도와 효율을 높일 수 있다.

#### 1. 서론

대부분의 시각 응용 프로그램 개발 도구는 복잡한 GUI 설계와 개발 시간의 단축을 통해 전체 개발 효율성을 증가시킬 수 있다. 그러나, 시각 응용 프로그램 개발자는 프로그램 엔진 모듈과 내부 로직 수행을 위한 인터페이스 개발을 모두 처리해야만 한다. 현재의 응용프로그램은 그 복잡성과 규모가 점점 증가하고 있다. 따라서, 시각 응용 프로그램 내부 로직의 복잡성뿐만 아니라, 사용자 인터페이스에 대한 개발 요구도 나날이 복잡해지고 있다. 그러므로, 단일 개발자가 개발 프로그램 전체를 설계하고, 요구 사항 변경이 발생했을 경우, 영향받는 모든 부분을 중횡(traverse)하며, 변경될 부분을 파악하고 처리해야 하기 때문에 개발 시간과 복잡함의 증가를 가져올 수 있다. 이것은 시각 프로그램 개발 도구가 시각 프로그램의 내부 로직과 GUI 부분을 기능별로 구별하여 여러 개발자가 공동 작업을 수행할 수 있는 기능을 제공하지 않는 데 있다.

본 논문은 여러 개발자가 협업을 통해 시각 프로그램의 내부 로직과 GUI 부분을 구분하여 자신의 업

무 영역에 맞는 개발에 참여할 수 있는 새로운 시각 프로그램 협업 개발 환경을 제안한다. 프로그램 개발을 위한 공동 개발 환경은 네트워크를 통한 자바용 프로그램 개발 툴이다. 제안된 시스템은 프로그램 개발자간 네트워크를 통해 즉각적인 의견 교환을 위해 통신 창을 제공한다.

#### 2. 본문

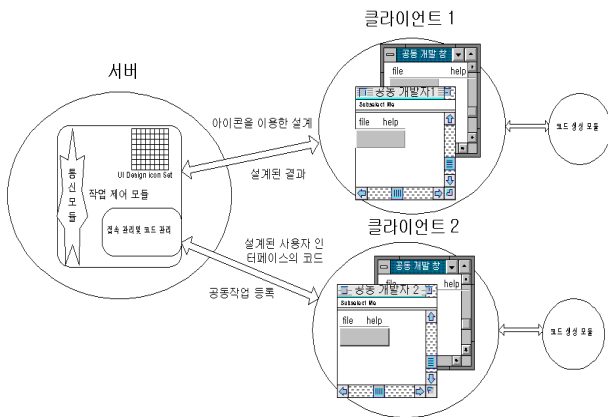
##### 2.1 프로그램 공동 개발 환경

프로그램 개발 초기 프로그램 개발 툴은 프로그래밍 언어 수준의 라이브러리(library)를 제공했다. 하지만, 더욱 시각적이고 편리한 사용자 인터페이스에 대한 요구는 시각 프로그램 개발환경을 발전 시켰다. CVS 환경(concurrent version system)은 공동의 저장소(repository)를 두고 네트워크에 접속해 공동으로 프로그램을 개발하기 위한 방법 중에 하나이다. 하지만, CVS를 이용한 공동 개발은 네트워크를 통해 다른 개발자의 개발 파일을

자신의 컴퓨터에 복사해 개발에 참고하고 다시 작성된 개발 파일을 서버에 복사하는 과정을 갖는다. 따라서, 즉각적인 공동의 개발 창에서 공동 작업을 할 수 있는 방법은 제공하지 않는다.

## 2.2 새로운 공동 개발 환경

본 논문에서 제시하는 새로운 시각 프로그램 공동 개발 환경은 네트워크 상으로 여러 사용자가 동시에 접근 가능한 공동 작업 창과 각 개발자 창을 제공하는 개발 모듈과 공동 개발의 신뢰성 있는 작업진행과 개발자간 동기 적인(synchronous)통신을 보장하기 위한 작업 제어 모듈, 그리고 개발된 사용자 인터페이스에 대한 코드 생성 모듈을 포함하는 통합 시스템이다. 본 논문에서 제안하는 공동 개발 환경의 전체적인 시스템 모델은 [그림 1]과 같다.

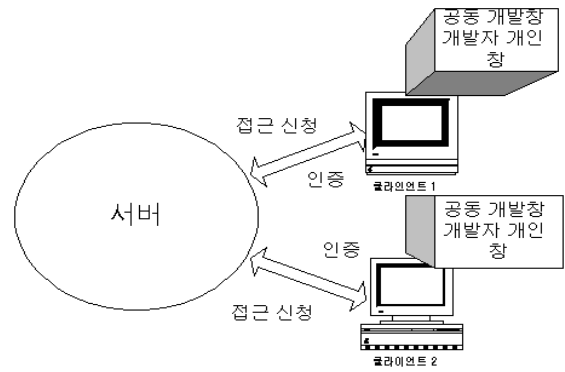


[그림 1] 시각 프로그램 공동 개발 시스템

제안하는 시스템은 네트워크를 통한 Server와 Client 구조를 가진다. 서버는 여러 개발자간의 공동 개발을 지원하기 위해 접근 제어와 작업 제어에 대한 책임을 지며, 클라이언트는 발생한 이벤트와 통신 메시지를 관리하기 위한 책임과 생성된 요소에 대한 코드 생성 및 관리를 책임진다. 서버는 스레드로 생성된 네트워크 관리 데몬들로 구성된다. 네트워크 관리 데몬들은 연결 전용 데몬과 클라이언트로부터 받은 메시지를 모든 참여자에게 브로드 캐스팅하기 위한 메시지 데몬, 그리고 작업의 일관성(concurrency)유지를 위한 컨커런스(concurrency) 유지 데몬으로 구성된다. 클라이언트 호스트는 해당 호스트에 있는 개발자에게 서버 호스트로의 연결을 위한 대화 창을 제공하고, 연결이 이루어지면 공동 개발 환경 초기 화면과 자신의 개발자 개인 창을 보여주며, 발생한 이벤트와 메시지를 서버에 보내는 역할을 수행하는 통신 데몬을 제공

한다. 개발 모듈은 공동 작업 창과 개발자 개인 창을 제공하고, 실제 공동 작업이 이루어지는 곳이다. 이곳에서의 모든 표현은 공동 작업이 이루어지는 과정 및 결과에 대한 투명성 보장을 위해 직접 조작(direct-manipulation)방법을 통한 시각적 표현이다.

제어모듈은 공동 개발에 참여하려는 개발자의 접근을 제어하기 위한 접근 제어 모듈과, 공동 개발 환경에서 이루어지는 모든 작업과정을 제어하기 위한 작업 제어 모듈, 그리고 각 개발자가 프로그램 개발 중 발생하는 의견을 전달할 수 있도록 통신을 제어하는 모듈로 구성된다. 제어모듈은 각 개발자의 설계과정에서 이루어지는 모든 입력에 대한 출력을 공동 개발 환경 화면에 직접 조작(direct-manipulation)방식을 통해 화면에 출력한다. 제어 모듈의 구성은 [그림 2]와 같다.



[그림 2] 클라이언트/서버 기반의 접근제어

## 2.3 메시지

시각 프로그램 공동 개발 환경을 통해 발생하는 메시지는 공동 개발자간 공동 개발 중에 서버와 지역 호스트간에 발생하는 이벤트 메시지와 공동 개발자간 협의를 위한 통신 메시지가 있다.

메시지 형태	개발자 고유 id	메시지 내용
--------	-----------	--------

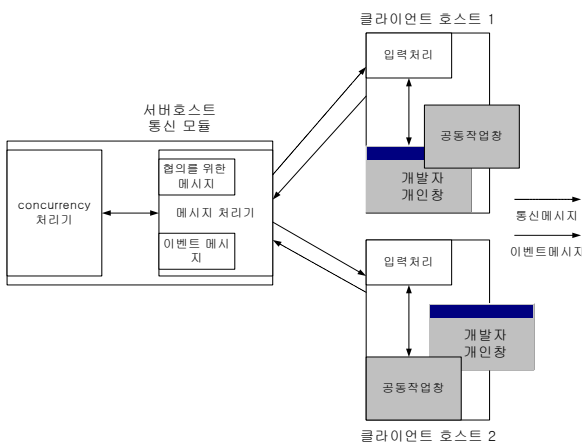
공동 개발자간 통신 메시지는 메시지 형태와 개발자 고유 id와 메시지 내용으로 구성된다. 메시지 형태는 메시지가 공동 개발자간 통신을 위한 메시지인지 시스템 메시지인지를 나타내 주기 위한 것이다. 개발자 고유 id는 개발자간 미리 정의된 id에 의한 구별을 위한 정보이다. 메시지 내용은 공동 개발자간 통신 창을 통해 전송되는 실제 메시지 내용이다.

메시지 형태	개발자 고유 id	프로젝트 이름	요소 종류	요소 속성
--------	-----------	---------	-------	-------

각 공동 참여자간 발생하는 이벤트 메시지 구조는 메시지 형태, 개발자 고유 id, 프로젝트 이름, 이벤트 요소 종류, 그리고 요소 속성으로 구성된다. 메시지 형태는 이벤트 메시지를 나타내 준다. 개발자 고유 id는 클라이언트 호스트에서 서버로 연결된 각 개발자의 고유 식별자로서 쓰이기 위한 정보이다. 프로젝트 이름은 하나의 개발 프로젝트 내에 여러 서브 프로젝트가 있을 때 해당 프로젝트를 나타내기 위한 정보이다. 작성된 요소의 종류는 생성하려는 사용자 인터페이스 요소 - 텍스트 영역, 텍스트 필드, 버튼, 리스트, 라벨 - 에 대한 정보이고, 요소 속성은 각 요소가 갖는 특성 - 크기, 색, 폰트 - 을 나타내는 정보이다.

### 2.4 데몬

클라이언트로부터의 메시지는 단순한 텍스트 통신 메시지와 설계 이벤트 메시지이다. 서버는 각 클라이언트로부터 받은 메시지를 이용해 응용 프로그램의 원본을 만든다. 원본의 복사본은 공동 개발자가 속한 클라이언트 호스트로 브로드캐스팅 방법을 이용해 전송된다. 이때 클라이언트가 가지는 사용자 인터페이스 응용 프로그램은 복사본이다. 이러한 일련의 과정을 처리하기 위해 서버와 클라이언트 호스트에는 스레드로 구현된 데몬이 존재한다. 다음의 [그림 3]은 데몬에 시각 프로그램 공동 개발 환경에 사용되는 데몬의 전체적인 구조이다.

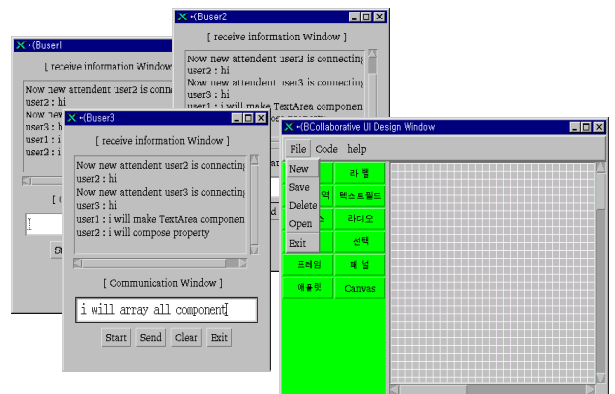


[그림 3] 공동 개발 환경의 데몬 개념도

서버의 메시지 입력 데몬은 클라이언트로부터의 입력 메시지를 받아 메시지 종류를 판단하여 일관성 (concurrency) 처리기로 전달하는 기능을 수행한다. 서버와 클라이언트간의 메시지 전송은 브로드캐스팅 방법을 사용하므로 여러 개발자간 공유된 설계 요소의 일관성 유지를 위해 이벤트 일관성 데몬이 사용된다. 일관성 처리기에서는 각 개발자가 발생한 이벤트 메시지를 조사하여 동일한 요소에 대한 이벤트일 경우 우선 처리 권한 모듈을 통해 순서를 정하고 순서에 따른 복사본을 만드는 일을 담당한다. 처리를 통한 메시지는 출력 처리 데몬을 통해 각 개발자에게 브로드캐스팅 되어진다. 클라이언트의 메시지 입력 데몬은 서버로부터의 입력 메시지를 받아 개발자의 개발자 개인 창에 보여준다. 출력 데몬은 각 개발자가 발생한 통신 메시지나 작업 이벤트 메시지를 서버로 보내는 기능을 한다.

### 2.5 공동 작업 창

개발 모듈은 공동 개발 창에서 이루어지는 실제적인 프로그램 설계 행위에 대해 자바 클래스로 작성된 사용자 인터페이스 도구 셋을 구성하는 클래스를 제공한다. 도구 셋은 아이콘형태로 제공된다. 초기 개발자가 서버로부터의 접근제어를 통해 지역 호스트에 연결하면, 지역 호스트의 통신 데몬은 개발자에게 공동 개발을 위한 공동 개발 창과 개발자 개인 창을 보여주게 된다. 다음 [그림 4]는 공동 개발 창과 통신 개인 창 제어 모듈에 의해 생성된 공동 개발 창과 통신 개인 창이다.



[그림 4] 공동 개발 창 과 통신 개인 창

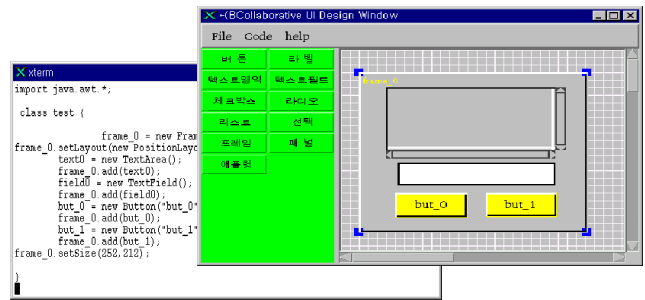
호출된 도구 클래스는 공동 작업 창에 생성되고, 위치 관리자(LayoutManager)에 의해서 그려진다.

## 2.6 코드생성

코드생성 모듈은 개발자들이 아이콘(icon)형태의 개발 도구 셋을 이용하여 공동 개발한 사용자 인터페이스 설계 요소에 대해 자바(java)코드를 생성한다. 각 개발 도구 셋 클래스에서 생성된 요소에 대한 스트링 정보를 하나의 프로젝트 정보로 만든다. 독립적인 사용이 가능하며, 다른 응용프로그램의 개발에도 사용될 수 있어서 코드의 재 사용성을 높인다.

## 3. 시스템 구현 및 실험

제안한 시각 프로그램 공동 개발 환경에 대한 실험은 간단한 형태의 인터페이스 요소를 포함하는 GUI 화면이고, 독립적인 프로그램 모듈로서의 GUI 화면과 해당하는 java 코드가 결과로서 출력된다. 공동 작업을 위해 2명의 개발자가 참여한다. 시각 프로그램을 개발하기 위해 한 개발자는 생성된 화면을 구성하는 인터페이스의 생성과 값을 설정한다. 또 다른 개발자는 [그림 4]와 같이 생성된 GUI 요소에 내부 엔진을 실행시키기 위한 메소드를 선언하고 정의한다.



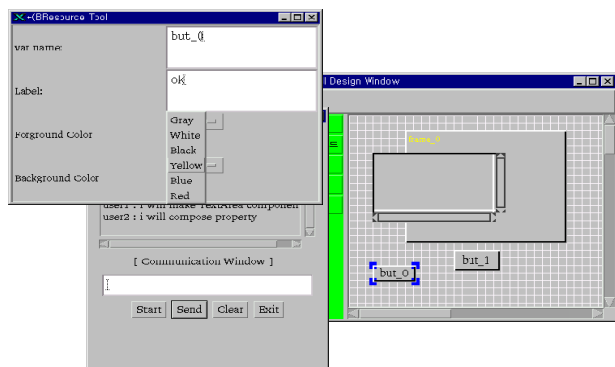
[그림 6] 생성된 인터페이스 요소와 자바 코드

## 4. 결론 및 향후 연구 방향

기존의 프로그램 개발 환경은 단일의 개발자가 모든 설계의 권한과 책임을 가진다. 따라서 규모가 크고 복잡한 프로그램의 개발에 있어 복잡한 인터페이스의 설계와 구현은 많은 부담이 된다. 본 논문은 여러 개발자가 프로그램을 공동으로 개발할 때, 공동 작업 공간에서 협의를 통해 프로그램 개발이 이루어질 수 있는 공동 개발 환경을 설계하고 구현했다. 제안한 공동 개발 환경은 여러 개발자간 즉각적인 의견 교환을 위해 통신 창을 제공하고, 개발자간 생성한 개발 코드에 대해 일관성을 유지하기 위한 데몬을 두었다. 따라서 복잡한 프로그램의 개발에 있어 개발을 분담하고 네트워크를 통해 즉각적으로 개발자간 협의가 가능하여 보다 작업의 효율과 정확성을 높일 수 있다. 앞으로 제안된 시스템을 네트워크 기반이 아니고 인터넷을 이용한 웹 기반으로 개발하여 보다 공간적, 시간적으로 활용도가 높고 제약성이 없는 공동 개발 환경으로 개발해야 할 것이다.

## 참고문헌

- [1] Brad A. Myers, State of the Art in User Interface Software Tools, Advances in Human-Computer Interaction, Volume 4, 1993
- [2] Brad A. Myers and Mary Beth Rosson, Survey on User Interface Programming. Humman Factors in Computing Systems, Proceedings SIGCHI'92, 1992, pp. 192-202.
- [3] Uwe Malinowski and Kumiyo Nakakoji, Using Computational Critics to Facilitate Long-term Collaboration in User Interface Design, Proceedings SIGCHI'95, 1995, pp. 385-392.



[그림 5] 화면 인터페이스 생성과 내부 로직 연결

통신 창을 통해 개발자간 의견 교환이 가능하고 협업을 통해 완성된 응용 프로그램의 인터페이스와 메소드 부분에 대해, 소스 코드가 자바 코드로 출력된다. 생성된 자바 소스 코드는 다른 요소를 설계하고 개발할 때, 라이브러리 형태로 사용될 수 있으므로, 코드의 재사용 면에서 활용도가 높아 개발 시간과 노력을 줄여 프로그램 개발 생산성을 높일 수 있게 된다. 다음의 [그림 6]은 시각 프로그램 공동 개발 환경을 통해 2명의 개발자가 설계하고 개발한 간단한 GUI를 포함하는 시각 프로그램의 화면과 자바 코드이다.