

시각 프로그래밍을 위한 공동 개발 환경

조용윤⁰, 박호병, 유재우

승실대학교 컴퓨터학과

{yycho⁰, r5me}@ss.ssu.ac.kr, cwyoo@comp.ssu.ac.kr

A Co-Development Environment for Visual Programming

Yong-Yoon Cho, Ho-Byung Park, Chae-Woo Yoo

School of Computing, Soongsil University

요 약

시각(Visual) 프로그램 개발은 프로그램 엔진 모듈에 대한 개발과 함께, 사용자 인터페이스 설계로 인해 더 많은 개발 시간과 노력을 필요로 한다. 빠르고 효과적인 시각 프로그램 개발을 위해 개발자의 능력과 업무 분야에 맞게 시각 프로그램의 엔진 부분과 사용자 인터페이스 코드를 나누어 개발하는 것이 바람직하다. 분업화된 시각 프로그램 개발은 동시에 여러 개발자의 참여를 가능하게 할 수 있다. 그러나, 오프라인 회의를 통한 공동 개발은 개발자간의 공간적, 시간적 독립성을 보장하지 못하며, 즉각적인 협의를 제한하기 때문에 비효율적이다. 본 논문은 개발자간 개발 독립성과 협동성을 동시에 만족하는 네트워크 기반의 시각 프로그램 공동 개발 환경을 제안한다. 제안하는 공동 개발 환경은 개발 시각 프로그램을 내부 로직과 사용자 인터페이스로 나누어 개발이 가능하며, 각 개발자의 독립적 개발을 보장하고, 개발자간 필요한 의견과 정보 교환을 지원하는 협업 프로그램 개발 환경이다.

1. 서론

시각화 응용프로그램 개발 환경은 단일 설계자에게 프로그램 처리 모듈과 모든 인터페이스에 대한 설계와 구현의 집중된 형태이다. 때문에, 단일 개발자는 프로그램 엔진 부분에 대한 알고리즘과 자료구조의 개발은 물론이고, 이것들을 사용자의 입력과 출력에 연결하기 위해 사용자 인터페이스 요소(component) 생성과 요소(component)의 행동(action) 정의 및 추가 코드 작성 등의 일을 혼자 해야만 한다. 현재의 응용프로그램은 그 복잡성과 규모가 점점 증가하고 있고, 그에 따른 사용자들의 사용자 인터페이스에 대한 요구도 날이 복잡해지고 있다. 그러므로 단일 개발자가 전체 설계를 처음부터 끝까지 모두 담당해야 한다는 것은 그 자체가 부담이 아닐 수 없다. 인터페이스를 구성하는 하나의 요소가 여러 개의 부 요소(sub component)로 구성되어 있고, 다시 반복적인(recursive)형태로 다른 요소를 포함하거나 다른 프로젝트와 연결되는 것과 같이 복잡성이 증가된 형태를 가질 때 이러한 부담은 증가된다. 또한 단일 개발자가 광범위한 설계에 대해 일관성을 유지하면서 전체적인 프로그램 개발 계획을 정확하게 숙지해야 한다는 사실은 실제로 어려운 일이다. 특히, 설계의 변경이 발생했을 경우, 단일 개발자는 영향받는 모든 부분을 중형(traverse)하며, 혼자 변경될 부분을 파악하고 처리해야 하기 때문에 개발 시간의 증가를 가져올 수 있다. 본 논문은 시각 프로그램 개발에

있어 여러 개발자가 능력과 작업 분야에 따라 분업 형태의 개발이 가능한 새로운 사용자 인터페이스 편집환경을 제안한다. 따라서, 큰 규모의 시각 프로그램 개발에 있어, 각 개발자는 개발 프로그램에 대한 전체 개발 책임과 작업 관리부담이 줄어들고, 여러 개발자가 공동 개발을 통해 빠른 시간 내에 사용자가 원하는 복잡하고 큰 규모의 응용 프로그램을 더욱 간단하고 편리한 방법으로 설계 할 수 있을 것이다. 제안된 시스템은 시각 프로그램 개발자간 협업을 위해 의견을 나눌 수 있는 통신 창을 제공하고 프로그램 설계를 의논하고 협업할 수 있도록 공동 작업 공간을 제공한다.

2. 본론

2.1 시각화 프로그램 개발 환경

프로그램 개발 초기 컴퓨터와 인간과의 대화는 대부분이 텍스트 기반에서 단순한 질의 응답의 형식을 띠었다. 때문에 초기 개발자들은 프로그래밍 언어 수준의 라이브러리(library)를 제공하는 개발 환경을 사용했다. 하지만 사용자들은 더욱 시각적이고 현실감 있는 인터페이스를 요구하게 되었고, 시각 프로그램은 이러한 개발환경을 발전 시켰다. CVS 환경(concurrent version system)은 공동의 저장

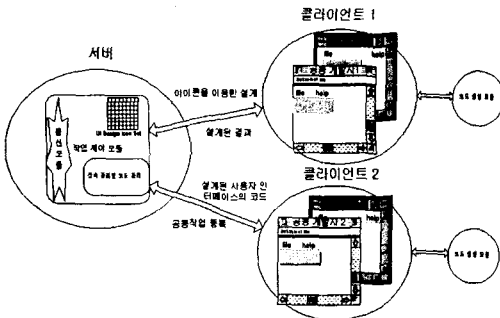
소(repository)를 두고 네트워크에 접속해 공동으로 프로그램을 개발하기 위한 방법이다. 하지만, CVS를 이용한 공동 개발은 네트워크를 통해 파일 버전을 관리하며, 다른 개발자의 개발 파일을 자신의 컴퓨터에 복사해 개발하는 과정을 갖는다. 따라서, 네트워크를 통해 공동의 개발 창에서 공동 작업을 할 수 있는 방법은 제공하지 않는다.

2.2 시스템 설계

본 논문에서 제시하는 새로운 시각 프로그램 공동 편집 환경은 자바(java)언어로 구현된 시각 프로그램을 위한 공동 편집 환경이다. 제안되는 시각 프로그램 공동 편집 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. 제안된 시스템은 다음과 같이 3가지 부분으로 나누어진다.

- 제어 모듈
- 작업 개발 모듈
- 코드 생성 모듈

본 논문에서 제안하는 공동 개발 환경의 전체적인 시스템 모델은 [그림 1]과 같다.



[그림 1] 프로그램 공동 개발 지원 시스템

서버는 여러 개발자간의 공동 개발을 지원하기 위해 접근 제어와 작업 제어에 대한 책임을 지며, 클라이언트는 발생된 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, 발생된 이벤트와 메시지를 서버에 보내는

역할을 수행하는 통신 데몬을 제공한다. 작업 제어 모듈은 사용자 인터페이스 공동 개발의 신뢰성과 보안성을 보장하기 위한 모듈이다. [2][3]

제어 모듈은 접근 제어 모듈, 공동 작업 창과 개발자 개인 창 생성 모듈 그리고 통신 모듈로 구성된다. 먼저 접근 제어 모듈은 공동 개발자가 초기 공동 작업장에 참여하려고 할 때, 특정 참여자의 참여 여부를 검증하는 기능을 가진다. 공동 작업 창과 개발자 개인 창 생성 모듈은 참여한 공동 개발자에게 공동 작업 창과 개인 창을 제공하며, 각 개발자가 생성하는 작업의 결과를 볼 수 있다.

개발 모듈은 실질적인 인터페이스를 공동 작업하기 위한 부분이다. 개발 모듈은 사용자 인터페이스 개발을 위한 아이콘(icon)형태의 도구(tools)를 제공한다. 이러한 도구는 자바(java)에서 제공하는 사용자 인터페이스 요소(component)를 나타내며, 자바(java) 클래스로 구현된다. 따라서 개발 모듈은 목표 사용자 인터페이스를 작성하기 위해 자바로 구현된 개발 도구 클래스들로 구성되어 있으며, 자체는 자바로 구현된 사용자 인터페이스 개발 틀의 형태를 갖는다.

마지막으로 통신 모듈은 개발자간 통신을 관리하고 제어하는 역할을 포함한다. 통신 모듈을 통해 개발자는 설계요소에 대한 독립적인 접근이 가능하게 된다. 따라서 각 개발자는 하나의 변경에 의해 자신의 책임 부분에 대해서만 조작의 책임을 가진다. 때문에 모든 부분에 대한 관찰의무와 설계 유지의 책임 부담은 없어지게 되어 진다.

2.3 공동 개발 메시지

공동 편집 환경에서 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 스트링 형태 메시지 형태와 개발자 고유 id와 메시지 내용으로 구성된다. 메시지 형태는 메시지가 공동 개발자간 통신을 위한 메시지인 것을 나타내 주기 위한 것이다. 개발자 고유 id는 서버와 클라이언트를 연결하기 위한 것이다.

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 식별자로써 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.

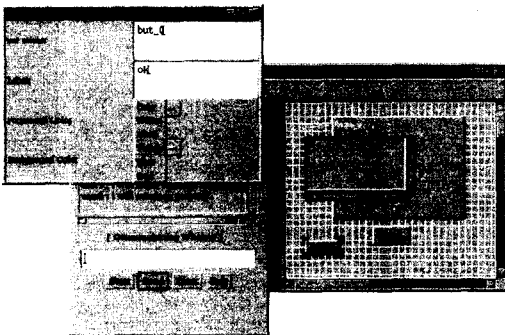
2.4 코드생성 모듈

코드생성 모듈은 개발자들이 아이콘(icon)형태의 개발 도구 셋을 이용하여 공동 개발한 사용자 인터페이스 설계 요소에 대해 자바(java)코드를 생성한다. 코드 생성기는 자신이 받은 복사된 스트링 정보를 이용해 각각 고유의 클래스 이름이 붙은 간단한 형태의 자바 코드를 생성한다. 서버는 각 개발자가 생성한 자바 코드를 유지하고 관리한다. 각 개발 도구 셋 클래스에서 생성된 요소에 대한 스트링 정보를 하나의 프로젝트 정보로 만든다. 이러한 정보는 설계된 사용자 인터페이스를 구성하는 모든 요소의 클래스 코드를 위한 정보이므로, 독립적인 사용이 가능하며, 다른 응용프로그램의 개발에도 사용될 수 있어서 코드의 재사용 성을 높인다.

3. 실험

본 논문에서 제안한 시각 프로그램 공동 개발 환경을 통한 간단한 사용자 인터페이스를 포함한 시각 프로그램의 개발 실험은 JDK 1.4버전에서 2명의 개발자가 참여하여 Sun ultra2와 Sun workstation server1000E를 이용하여 실시하였다.

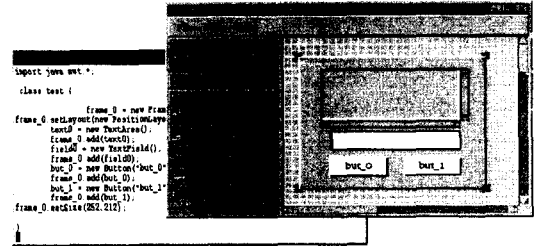
한 개발자는 개발할 시각 프로그램의 화면을 구성하는 인터페이스 요소를 생성하고, 속성 값 설정과 생성된 인터페이스 요소의 화면 위치를 담당한다. 또 다른 개발자는 [그림 2]와 같이 생성된 요소를 이용하는 프로그램의 엔진 부분 메서드 개발과 연결을 담당한다. 개발 중 발생하는 의견 교환을 위해 제안한 시각 프로그램 공동 개발 환경은 통신 창을 제공한다.



[그림 2] 화면 인터페이스 생성과 속성 부여

시각 프로그램을 위해 공동 편집 환경을 이용해 생성된 결과는 다음 [그림 3]과 같다. 전체 개발된 사용자 인터페이스와

각 요소의 메서드 부분에 대해, 소스 코드가 자바 코드로 출력된다.



[그림 3] 생성된 인터페이스 요소와 자바 코드

4. 결론 및 향후 연구 방향

본 논문은 여러 개발자가 시각 응용 프로그램을 공동 개발할 때, 공동 작업 공간에서 협의를 통해 프로그램 개발이 이루어질 수 있는 시각 프로그램 공동 편집 개발 환경을 설계하고 구현했다. 따라서 복잡한 시각 프로그램의 개발에 있어 개발을 분담하고 네트워크를 통해 즉각적으로 개발자간 협의가 가능하여 시각 프로그램 개발에 있어서 보다 작업의 효율과 정확성을 높일 수 있다. 또한, 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. 앞으로 제안된 시스템을 네트워크를 이용하지 않고 인터넷을 이용한 웹 기반으로 개발하여 보다 공간적, 시간적으로 활용도가 높고 제약성이 없는 시각 프로그램 공동 개발 환경으로 개발해야 할 것이다.

참고문헌

- [1] Chales Rich and Candace Sidner, Adding a Collaborative Agent to Graphical User Interfaces, ACM Symposium on User Interface Software Technology, 1996
- [2] Philip Isenhour, Clifford A. Shaffer, James Begole, Jeff Nielson, and Marc Abrams, "A Java-based Framework for Collaborative Interactive Modular Visualization Environments", Virginia University, TR-97-17, October 24, 1997.
- [3] Guinn, C. I, Mechanisms for Mixed-Initiative Human-Computer Collaborative Discourse, Proceedings 34th Annual Meeting of the Acl, 1996, pp.278-285.
- [4] James Begole, Craig A. Struble, and Clifford A. Shaffer, "Collaboration Transparency in Java through Event Broadcasting," IEEE Internet Computing, Vol. 1, No. 2, March - April 1997.
- [5] Mark D. Gross, Graphical Constraint sin CoDraw, Proceedings visual languages, 1992, pp. 81-87.