

# IIG: 인터페이스 상호작용 그래프

○  
최종명, 정용득, 신현경, 유재우  
승실대학교 컴퓨터학부

## IIG: Interface Interaction Graph

Jong-Myung Choi, Yong-Deg Jung, Hyun-Kyung Shin, Chae-Woo Yoo  
School of Computing, Soongsil Univ.

### 요 약

인터페이스 명세 방법들이 많이 개발되었지만, 현재 GUI 인터페이스 개발에는 인터페이스 명세 방법들은 사용되지 않고 인터페이스 디자인 중심의 개발 도구들이 주로 사용된다. 그러나, 현재 많이 사용되는 인터페이스 개발 도구들은 인터페이스 레이아웃을 기술하는 방법은 제공하지 않지만, 상호작용을 기술할 수 있는 방법은 제공해주지 못하고 있다. 이러한 문제점을 해결하기 위해 본 논문에서는 인터페이스 레이아웃과 상호작용을 동시에 지원할 수 있는 IIG(Interface Interaction Graph)와 IIG를 기반으로 한 Improve Builder 시스템을 소개한다.

### 1. 서론

1984년 Smith의 조사에 의하면 전체 코드에서 사용자 인터페이스는 30-35%를 차지하고 있으며[1], 또한 보다 최근인 1990년 MacIntyre가 조사한 바에 의하면 47-60%를 차지하는 것으로 나타났다[2]. 또한 비용면에서도 고해상도의 워크스테이션을 위한 인터페이스 개발에는 전체 비용의 약 29%가 소요되고, 그렇지 않은 경우에는 약 17%의 비용이 소모되는 것으로 조사됐다[3].

인터페이스의 중요성 때문에 인터페이스 명세에 관한 연구가 많이 이루어졌는데 대표적인 것으로는 문법, 태스크 기술 언어, 상태전이도, 상태 차트, IRG(Interface Representation Graph) [5], 규칙 기반 시스템 등의 방법들이 개발되었다. 그러나, 이러한 인터페이스 명세 기술 방법은 실제계에서는 많이 사용되지 않고, GUI를 디자인 할 수 있는 HyperCard, Visual Basic 등의 인터페이스 개발 도구들이 많이 사용된다. 이러한 도구들을 사용하는 경우에 50% ~ 500%에 달하는 생산성 향상을 가져오는 것으로 알려졌다[4].

현재 사용되는 인터페이스 개발 도구들은 인터페이스 컴포넌트의 레이아웃/크기/색/폰트 등을 결정하거나 인터페이스 컴포넌트의 속성 값을 설정해주는 기능은 제공하지만, 인터페이스와 사용자간의 상호작용을 기술할 수 있는 기능은 제공하지 못하고 있다. 인터페이스

디자인 기능과 더불어 인터페이스와 사용자간의 상호작용을 기술할 수 있다면 인터페이스 개발을 더 쉬워지고 생산성은 향상될 것이다. 본 논문에서는 인터페이스 디자인과 상호작용을 동시에 지원할 수 있는 IIG와 현재 개발중인 IIG 기반의 Improve Builder 시스템을 소개한다.

### 2. IIG(Interface Interaction Graph)

IIG는 GUI 인터페이스의 레이아웃과 인터페이스와 사용자간의 상호작용을 표현한다. IIG가 IRG와 다른 점은 IIG는 인터페이스의 형태/레이아웃과 상호작용을 동시에 지원한다는 것이다. IRG는 인터페이스 컴포넌트의 레이아웃과 컴포넌트의 특성을 기술하지 못한다는 단점이 있다. 반대로 Delphi나 Visual Basic같이 일반적으로 사용되는 인터페이스 개발 도구는 레이아웃과 컴포넌트의 특성은 기술하지만 상호작용을 기술하지 못한다. 이 두 종류의 시스템이 가지고 있는 단점을 보완한 것이 IIG이다.

IIG에는 4가지 형태의 노드가 있다.

- 위젯노드(Widget Node)
- 슈도 위젯노드(Pseudo-Widget Node)
- 액션노드(Action Node)

• 시작노드(Start Node)

2.1 위질노드

위질노드는 GUI 위질을 표현하기 위한 것으로 프레임, 버튼, 다이얼로그, 텍스트 영역 등 많은 종류가 있으며 각 위질의 형태와 유사한 모양으로 표현된다. 그림1은 버튼을 위한 위질노드의 모습이다

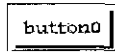


그림 1 위질노드

위질노드는 위질의 특성에 따라 기본적으로 2가지 형태로 구분할 수 있다. 버튼, 라벨, 리스트와 같은 프리미티브 위질이 있는 반면에 프레임, 윈도우같이 다른 위질을 포함할 수 있는 컨테이너 위질도 있다. IIG에서는 상태차트(Statechart)의 부모노드나 IRG의 그룹 노드(grouping of nodes) 같은 다른 노드를 포함하는 노드를 별도로 가지고 있지 않다.

2.2 슈도 위질노드

슈도 위질노드는 위질노드를 일시적으로 그룹화하기 위해 사용되는 노드이다. 슈도 위질노드는 주로 인터페이스를 구성할 때 화면에 직접 그림으로 표현하기 어려운 내용을 표현할 때 사용된다. 예를 들면, 메뉴바에서 메뉴를 구성한다던가, 팝업메뉴에서 팝업 메뉴를 구성할 때 슈도 위질노드는 메뉴를 포함하는 노드로 사용된다. 그림2는 슈도 위질노드의 모습이다.

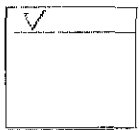


그림 2 슈도 위질노드

2.3 액션노드

액션노드는 어떤 작업을 수행하기 위한 코드가 기술된 노드이다. 이벤트 발생에 의해 액션노드로 상태가 진이되빈 노드에 기술된 코드가 수행된다. 액션노드와 유사한 개념으로 IRG에서는 공유 데이터 노드(Shared Data Node)를 제공하고 있다. 공유 데이터 노드는 인터페이스와 응용프로그램간의 통신을 위한 목적으로 제공되어 졌다. IIG에서 액션노드는 공유 데이터 노드의 개념을 확장하여 데이터뿐만 아니라 코드까지 기술할 수 있도록 하였다. 그림3은 액션노드의 모습이다.



그림 3 액션노드

2.4 시작노드

시작노드는 인터페이스의 시작을 알려주는 노드로서 액션노드의 일종이다. 시작노드는 크게 2가지 역할을 수행한다. 첫 번째는 인터페이스의 시작을 알림으로서 GUI 인터페이스의 처음 화면을 알려준다. 두 번째 역할로는 인터페이스와 응용프로그램의 결합을 위해 초기화 작업을 수행한다. 시작노드는 액션노드의 일종이기 때문에 코드를 기술할 수 있는 것이다. 그림4는 시작노드의 모습이다



그림 4 시작노드

2.5 아크

IIG에는 한 종류의 아크가 있고, 아크는 3개의 속성을 표현할 수 있다. 아크에는 이벤트와 컨스트레인트를 기술할 수 있다. 이벤트와 컨스트레인트는 각각 생략할 수 있다. 이벤트가 생략된 경우에는 아크의 꼬리에 있는 노드에서 디폴트 이벤트가 발생한 것을 의미한다. 컨스트레인트가 생략된 경우에는 컨스트레인트가 없는 것을 의미한다. 아크는 디펜던트 객체의 modality와 소스 객체와 디펜던트 객체의 가시성에 관한 속성을 표현할 수 있다. 아크의 꼬리에 있는 원이 적색이면 modalesis 이고, 적색이면 modal을 의미한다. 아크의 꼬리에 있는 원이 색이 채워져 있으면 소스 객체가 보이는 상태이고, 원안에 색이 채워져 있지 않으면 이벤트가 발생하는 경우에 소스 객체가 보이지 않게 된다. 유사하게 아크의 머리부분에 있는 원이 없는 경우에 디펜던트 객체는 이벤트가 발생하면 보여지게 되고, 원이 있으면 이벤트가 발생하면 디펜던트 객체가 안보이게 된다. 그림5는 일반적인 아크의 형태이다.

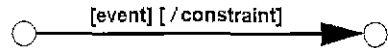


그림 5 아크

그림6은 메뉴아이템을 마우스로 클릭해서 파일 선택 다이얼로그가 화면에 나타나도록 하는 것을 IIG로 표현한 것이다. 메뉴아이템은 마우스 클릭이 디폴트 이벤트로 등록되어 있기 때문에 아크에 이벤트를 기술하지 않았다. 컨스트레인트는 없는 상태이고, 디펜던트 객체인 파일 선택 다이얼로그가 화면에 보여지도록 하기 위해서 아크의 머리부분에 원이 없다. 아크의 꼬리에 있는 원은 적색이고 원의 안이 채워져 있지 않기 때문에 소스 객체는 이벤트가 발생하면 화면에서 사라지고, 파일 선택 다이얼로그는 modal 형태로 사용된다

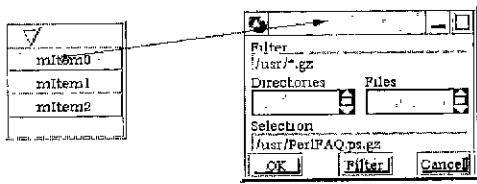


그림 6 IIG 아크 사용 예

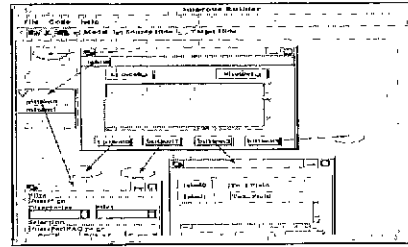


그림 8 Improve Builder의 그래픽 에디터

### 3. Improve Builder

Improve Builder는 IIG를 기반으로 현재 개발중인 인터페이스 개발 도구이다. Improve Builder는 IIG를 기반으로 한다는 가장 큰 특징이외에 코드 생성기를 등록만 하면 여러 가지 언어로 코드를 생성할 수 있다는 특징이 있다. 따라서 확장성 및 재사용성이 매우 뛰어나다. Improve Builder는 다음 그림7과 같은 구조로 되어 있다. 컴포넌트 프레임은 사용할 노드를 선택할 수 있는 윈도우이고, 그래픽 에디터는 인터페이스와 상호작용을 디자인할 수 있는 윈도우이다. 텍스트 에디터는 직접 코딩할 필요가 있는 경우에 사용된다. 그래픽 에디터에서 표현된 인터페이스 그림들은 모두 트리 핸들러에 의해 트리 형식으로 변환 관리된다. 코드 생성기는 트리 핸들러에 등록을 하고 코드를 생성할 수 있으며, 동적으로 코드 생성기를 등록할 수 있다.

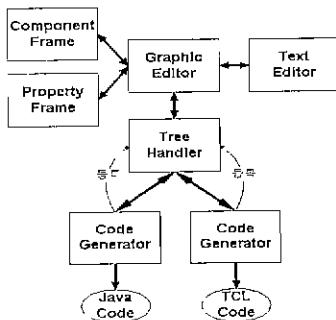


그림 7. Improve Builder 구성

그림8은 뉴스 그룹에서 뉴스를 읽고 포스팅할 수 있는 프로그램의 인터페이스를 Improve Builder를 이용해서 표현한 예이다.

### 4. 결론 및 향후 연구

본 논문에서는 인터페이스를 디자인하고 상호작용을 기술할 수 있는 IIG와 IIG를 바탕으로한 Improve Builder를 소개하였다. IIG는 현재 인터페이스 개발 도구에서 지원하기 어려운 상호작용을 사용자가 이해하기 쉽게 표현할 수 있다는 장점을 가지고 있다. Improve Builder는 IIG를 바탕으로 인터페이스 상호작용을 쉽게 표현해서 인터페이스 개발이 더 용이하고, 코드 생성기를 동적으로 등록함으로써 많은 다른 언어로 코드를 생성할 수 있다. 앞으로 IIG를 개선해나가는 연구와 함께 Improve Builder를 계속 개발할 예정이다.

### 5. 참고 문헌

- [1] Smith S. L, Mosier J. N., "Design guidelines for the user interface for computer-based information systems", Bedford, MA The MITRE Corp., 1981.
- [2] MacIntyre F., Estep K.W., Sieburth J. M., "Cost of user-friendly programming", *Journal of Forth Application and Research*, 6(2), pp.103-115, 1990
- [3] Rosenberg, D., "Cost-benefit analysis for corporate user interfaces standards: What price to pay for a consistent look and feel", in *Coordinating user interfaces for consistency*, New York Academic Press, pp.21-34, 1989.
- [4] Ben Shneiderman, *Designing the User Interface Strategies for Effective Human-Computer Interaction*, Addison Wesley, 1998.
- [5] Christopher Rouff, *Specification and Rapid Prototyping of User Interfaces*, Ph.D Thesis, Univ of Southern California, 1991