

VIOLA : 웹 응용 프로그램 개발을 위한 자료흐름 방식의 시각 프로그래밍 도구

조 창 식[†] · 신 규 상^{††} · 마 평 수^{††}

요 약

본 논문에서는 VIOLA(Visual Object-oriented Language for Multimedia Applications)의 시각 언어 프레임워크를 제안한다. VIOLA는 웹 응용 프로그램을 자동으로 생성시켜주는 도구이며, 사용자가 정확한 HTML 태그나 CGI 프로그래밍을 모르더라도 자동으로 고수준의 웹 응용 프로그램을 작성할 수 있게 해준다. VIOLA에서는 사용자 접속 설계 부분과 프로그램의 논리 설계 부분이 자료흐름 방식에 의해 통합되어 나타난다. 사용자 접속 설계 부분은 직접 조작과 위치위를 지원하고, 프로그램 논리 설계 부분은 추상화된 아이콘과 사용자 접속과의 연결을 위한 자료흐름 방식을 사용한다. 프로그램 논리는 도구에서 정의된 클래스를 재사용하여 결합함으로써 자동으로 CGI 프로그램으로 생성되어진다. 이러한 방법은 직관적이고 사용하기 쉬워서 초보자라도 쉽고 빠르게 고수준의 웹 응용 프로그램을 생성할 수 있다. 또한 웹 문서의 생성과 저장에 필요한 일련의 기능들이 포함되어 있다. 생성된 CGI 프로그램은 Java로 구현되기 때문에 HTML 문서와 더불어 플랫폼에 독립적이다.

VIOLA : An Iconic Data-flow Visual Language for Web Applications

Chang-Sik Cho[†] · Gyu-Sang Shin^{††} · Pyeong-Soo Mah^{††}

ABSTRACT

In this paper, we propose the framework of a visual language named VIOLA(Visual Object-oriented Language for Multimedia Applications), which is developed for web authoring. VIOLA supports the user to construct HTML documents and CGI programs even if he/she doesn't know exact HTML tags and CGI programming. In VIOLA, a user interface building part and a programming logic building part are combined by the data-flow model. The user interface building part provides direct manipulation and a WYSIWYG interface, and the program logic building part provides more intuitive interface by using predefined classes and the data-flow model. Thus, even a novice user can construct sophisticated applications by simply moving or clicking the mouse.

Several classes which generate CGI codes are predefined, and they are represented with highly abstracted visual components. By reusing predefined classes, CGI codes are automatically generated. In VIOLA, all major steps in building and publishing web applications are integrated. Moreover, the generated codes are platform independent because they are generated with HTML documents and Java CGI programs.

[†] 정 회 원 : 한국전자통신연구원 연구원
^{††} 정 회 원 : 한국전자통신연구원 책임연구원
논문접수 : 1998년 3월 27일, 심사완료 : 1999년 1월 29일

1. 서 론

인터넷과 웹의 발전에 따라 기존의 클라이언트/서버 응용 프로그램이 웹 상에서 개발되고 있다. 웹을 표준 인터페이스로 사용함으로써 통일된 사용자 인터페이스를 제공할 수 있으며, 개방성, 이식성, 유지보수 면에서 장점을 가지게 된다. HTML(HyperText Markup Language) 문서는 정보의 교환 수단으로서 중요한 의미를 가지게 되었으며, CGI(Common Gateway Interface)를 이용한 고수준의 응용 프로그램 개발이 증가하고 있다. 그러나, HTML 문서의 편집은 까다로운 태그의 학습을 요구하며, CGI 프로그램의 개발은 일반 사용자에게 어려운 프로그래밍 기술을 요구한다. 따라서 개발자들이 웹 응용 프로그램을 작성하는 과정은 여전히 많은 시간과 노력을 요구하게 된다[1].

많은 웹 저작 지원 도구들이 개발되었는데, 이러한 도구들은 크게 두 가지 범주로 나누어질 수 있다. 첫 번째는 단순한 웹 문서의 저작을 위한 도구이고, 두 번째는 데이터베이스 접속과 같은 고수준의 웹 응용 프로그램을 개발하는 도구이다. 전자는 HTML 문서의 편집에 중점을 두며, 태그의 기능을 처리하는 다양한 도구를 아이콘의 형태로 제공함으로써 HTML 페이지 개발 환경을 제공한다[2,3,4]. 또한 위치위(What You See Is What You Get) 기능을 제공함으로써 HTML 태그의 사용을 사용자에게 숨겨 주며, 내용 관리나 전체 사이트에 대한 레이아웃 설계 기능을 제공한다. 그러나 이러한 도구들은 CGI 프로그램의 생성을 지원하지 못한다. 따라서 폼을 사용하여 데이터베이스를 검색하는 응용 프로그램을 생성하기 위해서는 도구와 별도로 텍스트 입력에 의한 저수준의 프로그래밍을 요구하게 된다. 이러한 저수준의 프로그래밍은 개발자들에게 어려움을 겪게 한다.

후자는 미리 정의된 템플릿과 위저드를 제공하여 데이터베이스 질의문을 생성하고 수행시키며 질의 결과를 포맷팅하게 된다[5,6,7]. 이러한 일련의 과정은 도구가 제공하는 다이어로그와 그에 대한 입력으로 이루어지는데, 입력으로는 사용해야 될 템플릿 및 템플릿에 대한 매개변수이다. 입력된 내용에 따라 CGI 프로그램이 자동으로 생성되어지고, 생성된 프로그램은 데이터베이스 처리를 자동으로 수행하게 된다. 그러나 이러한 도구들은 사용하기에 너무 복잡하며, HTML 문서의 포맷팅과 프로그램의 논리와의 연결 과정을 위

해 변수를 사용하게 함으로써 복잡한 변수에 대한 관리를 사용자에게 요구한다.

본 논문에서는 시각 프로그래밍 도구인 VIOLA(Visual Object-oriented Language for Multimedia Applications)의 프레임워크를 제안한다. 본 논문에서 제안하는 VIOLA 도구는 기존의 도구와는 달리 사용자 접속 설계 부분과 프로그래밍 논리의 설계 부분을 자료흐름 방식에 의해 결합함으로써 HTML 문서의 편집 기능과 CGI 프로그램의 생성 기능을 하나의 도구에서 지원하도록 하고 있다. 따라서 사용자는 HTML 태그의 문법이나 CGI 프로그래밍 기법 등을 모르더라도 쉽게 원하는 웹 응용 프로그램을 생성할 수 있다.

사용자 접속 설계는 직접 조작(Direct Manipulation)과 위치위 편집 기능을 제공하고, 프로그래밍 논리 부분은 추상화된 아이콘과 사용자 접속과의 연결을 위한 자료흐름 방식을 사용함으로써 직관적인 인터페이스를 제공한다. 위치 정보나 시각 아이콘의 속성에 대한 정보들은 해당되는 HTML 태그나 CGI 프로그램으로 자동으로 생성되므로 초보자라도 쉽고 빠르게 고수준의 웹 응용 프로그램을 생성할 수 있게 해 준다. HTML 태그나 CGI 코드를 자동으로 생성하기 위한 클래스들이 도구에서 정의되어 있다. 정의된 클래스를 재사용하여 서로 결합함으로써 데이터베이스 질의와 멀티미디어 재생 기능 등이 쉽게 구현될 수 있다.

VIOLA에서는 웹 응용 프로그램의 생성과 공표(publishing)에 필요한 일련의 과정이 통합되어 있다. 시각적으로 생성된 프로그램은 코드의 자동 생성, 코드의 컴파일, 코드의 서버 이동 과정을 통하여 즉시 실행 가능하다. 또한 생성된 CGI 프로그램은 Java로 구현되었기 때문에 HTML과 더불어 플랫폼에 독립적이다.

2. 시각 프레임워크

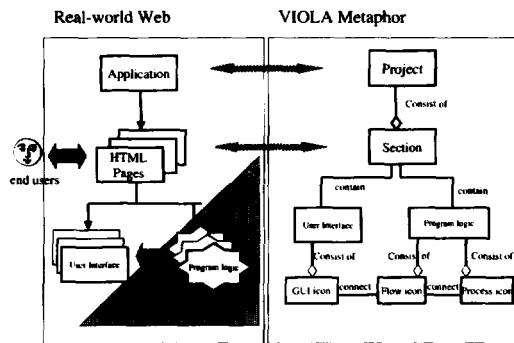
본 절에서는 시각 메타포, 아이콘과 수입 아이콘, 제약조건 등 VIOLA의 프레임워크에 대하여 설명한다.

2.1 시각 메타포

VIOLA는 아이콘 기반 자료흐름 방식의 시각 언어이다. 즉 모든 요소들이 아이콘으로 표시되며, 프로그램의 논리를 표현하기 위해 자료흐름 방식이 사용된다. 자료흐름 방식은 기능을 나타내는 노드와 기능들

사이의 자료의 흐름을 나타내는 선으로 프로그램을 표현한다. 노드로 들어가는 선은 기능에 대한 입력이 되고, 노드에서 나오는 선은 수행 결과에 대한 출력이 된다[8]. 자료흐름 방식에 의한 기존의 도구들로는 Prograph[9], VISTA[9], HI-VISUAL[10], VisualAge [11] 등이 있으며, 직관적이고 사용하기 쉬워서 많이 쓰여진다.

시각 언어는 응용 분야의 특성을 도구에 반영하여야 한다. VIOLA는 페이지 단위로 보여지는 웹의 특성을 반영하기 위하여 프로젝트-섹션-아이콘 메타포를 사용한다. (그림 1)은 실제 웹 환경과 시각 메타포 사이의 관계를 보여주는데, 메타포는 OMT[12] 객체 모델로 표현되었다.



(그림 1) 프로젝트-섹션-아이콘 메타포
(Fig. 1) Project-section-icon metaphor

사용자 관점에서 웹 응용 프로그램은 HTML 문서나 CGI 프로그램에 상관없이 실행시에는 페이지들의 나열로 보여진다. 전체 웹 응용 프로그램은 프로젝트에 해당되고, 각각의 페이지는 섹션에 해당된다. HTML 태그로만 이루어진 웹 페이지는 사용자 접속이 섹션과 대응되고, 데이터베이스 검색 결과의 출력 등과 같은 논리가 수행된 결과로 나타나는 화면은 사용자 접속과 더불어 프로그램의 논리도 섹션에 포함되어야 한다. 따라서 섹션은 실행시 보여지는 프로그램의 논리 부분도 포함하게 된다. 물론 사용자 접속만을 포함하는 페이지도 하나의 섹션으로 구성될 수 있다. 만일 섹션이 논리 부분을 포함하고 있다면 사용자 접속 부분을 처리하면서 프로그램 논리를 기술하는 CGI 프로그램이 생성되어진다.

아이콘은 섹션을 구성하는 요소들이다. 아이콘은 사

용자 접속을 취급하는 GUI 아이콘, 프로그램 논리를 위한 프로세스 아이콘, 사용자 접속과 프로그램 논리의 연결을 위한 흐름 아이콘으로 구성된다. 프로젝트-섹션-아이콘 메타포를 사용하여 사용자 접속을 중심으로 응용 프로그램을 작성하게 함으로써, 프로그램 논리가 직관적으로 나타나게 됨으로써 사용자가 쉽게 프로그래밍을 할 수 있다.

2.2 아이콘

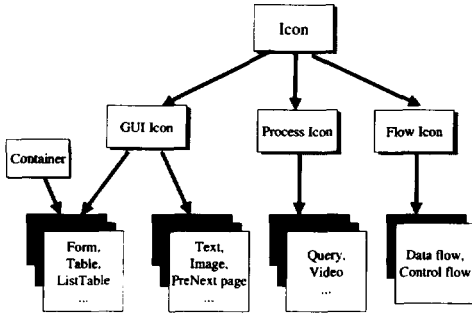
아이콘은 개발자에 의해 취급되는 최소 단위이다. 각 아이콘은 아이콘의 모양과 속성, 소스코드 생성 규칙을 가지고 있다. 아이콘의 모양은 비트맵이나 윈도우 컨트롤 등 화면 상의 표현 방식과 위치 정보를 포함하고, 속성은 상태를 나타내는 에트리뷰트, 아이콘이 인식할 수 있는 사건을 기술하는 이벤트, 특정 이벤트에 대해 동작할 수 있는 오퍼레이션을 정의한다. 아이콘의 에트리뷰트 값은 초기값으로 설정되고, 다이얼로그를 이용하여 정적으로 변경되거나, 흐름 아이콘의 연결에 의해 실행 시간에 값이 결정되는 동적인 변경이 가능하다.

(그림 2)는 아이콘의 계층구조를 나타낸다. 아이콘은 GUI 아이콘, 프로세스 아이콘, 흐름아이콘으로 구성된다. GUI 아이콘은 웹 사용자 접속을 위해서 사용되며, 아이콘의 모양은 웹 브라우저에서 나타나는 모양과 동일하게 표현된다. GUI 아이콘은 폼 아이콘, 테이블 아이콘, 확인(submit) 아이콘 등이 있으며, HTML 태그와 일대일로 대응된다. 또한 ListTable 아이콘, PreNextPage 아이콘 등이 같이 동적 자료의 포매팅을 위한 아이콘이 정의되었으며, 이 아이콘들은 임시 템플릿 코드를 생성하게 되고 CGI 코드 생성에 필요한 정보가 된다. ListTable 아이콘은 실행된 질의의 결과를 웹 브라우저에 포매팅하기 위한 아이콘이며, PreNextPage 아이콘은 질의 검색 결과가 설정된 페이지 크기를 넘어갈 때 이전 페이지나 다음 페이지로의 이동을 지원하는 아이콘이다.

프로세스 아이콘은 프로그램의 논리를 표현하기 위해 사용되는 아이콘이다. 질의 아이콘은 데이터베이스 검색을 위해 필요한 아이콘이고, 비디오 아이콘은 멀티미디어 데이터의 재생을 위한 아이콘이다.

흐름 아이콘은 방향성 있는 선으로서 두개의 아이콘 사이에 정보의 교환이 필요할 때 사용된다. 연결은 두 아이콘 사이의 에트리뷰트의 교환을 나타내는 자료

의 흐름이거나 한 아이콘의 이벤트가 다른 아이콘의 오퍼레이션을 호출하는 제어의 흐름이다.



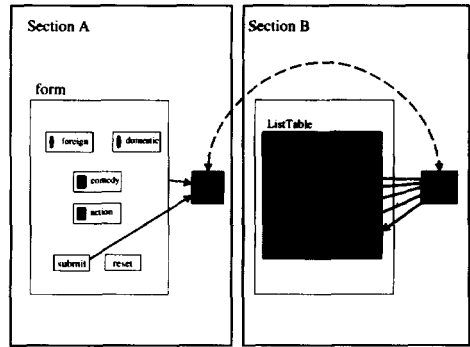
(그림 2) 아이콘 계층구조
(Fig. 2) Icon Hierarchy

2.3 수입 아이콘

사용자는 이차원 화면상에 아이콘을 배열하고 배열된 아이콘을 연결시킴으로써 응용 프로그램을 생성하게 된다. 아이콘 사이의 정보 흐름을 표현하기 위하여 흐름 아이콘이 사용되는데, 이는 단지 한 섹션 내에서만 유효하다. 그러나 실제로는 사용자 접속 화면과 사용자 접속 화면 사이에 정보의 공유가 필요한 경우가 있다. 예를 들어 사용자가 폼을 생성하고 폼 데이터를 처리하고자 하면 CGI 프로그램에 입력되는 매개변수가 필요하게 된다. 섹션이 웹 페이지에 대한 사용자 접속 설계 단위로 구성되기 때문에 CGI 프로그램에 대한 사용자 접속은 별도의 섹션에서 기술되게 된다. CGI를 처리하는 섹션에서 프로그래머는 인자가 전달되는 것을 알아야 한다. 이 개념을 지원하기 위하여 VIOLA에서는 수입 아이콘(Import icon)을 제공한다. 수입 아이콘은 섹션과 위치 정보를 제외하고는 원래 아이콘과 동일하며 원래 아이콘의 속성을 그대로 사용한다. 원래 아이콘의 속성이 바뀌면 수입된 아이콘의 속성도 자동으로 바뀌게 되며, 수입 아이콘의 속성이 변화하면 원래 아이콘의 속성도 변화된다.

(그림 3)은 수입 아이콘의 사용 예를 보여준다. 섹션 A에서 폼은 두개의 체크박스, 두개의 라디오박스, 확인 버튼, 재설정 버튼을 가지고 있다. 또한 데이터베이스 검색을 위한 질의 아이콘이 프로세스 아이콘 영역에 나타나고 있다. 확인 아이콘이 선택되면 데이터베이스 검색을 위한 CGI 프로그램이 호출되며, 이는 다음 섹션에서 사용자 접속과 더불어 처리된다. 폼에

서 사용자가 입력한 값을 얻기 위하여 폼의 인자 값이 CGI 프로그램의 인자로 전달되어야 한다. 섹션 B에서 수입한 질의 아이콘은 섹션 A에 있는 질의 아이콘의 속성을 공유하게 된다. 따라서 폼에 대한 데이터베이스 처리를 위해서 사용자는 섹션 A와 B 어디에서나 질의 생성기를 구동시킬 수 있고, 생성된 SQL 질의문을 사용할 수 있다.



(그림 3) 수입 아이콘
(Fig. 3) Import icon

연결 아이콘이 폼 아이콘과 질의 아이콘을 연결하는 시점에 체크박스와 라디오박스의 NAME 변수가 추출된다. 섹션 B에서 수입된 질의 아이콘은 질의 아이콘의 속성을 사용하여 SQL 질의문을 생성하게 되고, 자료흐름을 이용하여 결과를 ListTable 아이콘에 출력하게 된다. 질의 생성기에 의해 선택된 데이터베이스 필드 값은 질의 아이콘의 에트리뷰트 속성으로 추가되고, 그 에트리뷰트는 ListTable과 연결하는데 사용된다.

2.4 제약조건

VIOLA에서는 일치성 검사 기능이 편집 단계에서 대화적으로 수행된다. 따라서 사용자가 아이콘을 유효하지 않는 영역에 위치시키거나, 연결 불가능한 두 아이콘을 연결시키는 행동을 제한한다. 아이콘의 위치나 아이콘의 연결에 대한 규칙들은 제약조건으로 관리된다. 제약조건이 만족되지 않는 입력은 편집 단계에서 무시된다.

VIOLA의 제약조건에는 공간 제약조건, 연결 제약조건, 편집 제약조건이 있다. 공간 제약조건은 아이콘 사이의 포함 관계를 정의한다. 아이콘의 포함관계에 대한 아이콘의 계층구조가 (그림 2)에 나타나 있다. 예를 들어 확인(submit) 아이콘은 폼 아이콘의 영역 안

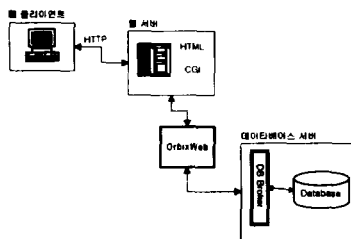
에 포함되어야 한다. 연결 제약조건은 두 아이콘 간의 연결 가능성을 제한하며, 연결 가능한 아이콘의 쌍이 정의되어 있다. 예로 폼 아이콘의 확인 버튼은 질의 아이콘의 검색 오퍼레이션을 수행시킬 수 있다. 그러나 HTML의 구성요소가 다른 HTML 구성 요소를 제어하는 일이 없기 때문에, GUI 아이콘은 다른 GUI 아이콘을 제어 흐름에 의하여 연결할 수 없다. 편집 제약조건은 한 아이콘의 편집에 대한 다른 아이콘의 영향에 대하여 기술한다. 만약 폼 아이콘이 삭제된다면 포함된 모든 폼 아이콘의 구성 요소들도 자동으로 삭제되며, 이동하게 되면 포함된 구성 요소들도 폼 아이콘의 이동 거리에 비례해서 이동하게 된다.

3. 구현 및 결과

본 절에서는 앞 절에서 제안한 프레임워크에 기초하여 구현에 관계된 요소에 대하여 기술한다. 생성되는 응용 프로그램의 구조, 소스코드의 생성 방법, 시스템 구조 및 예제 프로그램에 대하여 기술한다.

3.1 응용프로그램의 운용 환경

VIOLA가 생성하는 응용 프로그램의 운용 환경은 (그림 4)와 같다. 개발된 응용 프로그램은 클라이언트에서 웹 브라우저에 의해 수행되며, HTTP 프로토콜을 사용하여 웹 서버와 통신한다. 웹을 사용자 접속으로 사용하므로 플랫폼에 독립적인 응용 프로그램의 생성이 가능하며, 사용 편의성을 증대시킨다. 데이터베이스 서버의 기능을 처리하는 데이터베이스 브로커는 분산 객체 표준인 CORBA[13,14,15] 위에서 연동되며, 데이터베이스에 대한 접속 및 SQL 질의 처리를 지원한다. 데이터베이스 브로커는 OrbixWeb[15]을 사용하여 Java CGI 프로그램과 통신하게 된다. Java 프로그램은 IDL



(그림 4) 응용 프로그램의 운용 환경
(Fig. 4) Architecture of target applications

에서 정의된 객체들의 메소드를 호출하게 되고, 분산 객체들은 CGI 프로그램을 대신해 응용 프로그램을 처리하고 결과를 CGI 프로그램에게 돌려준다. CGI 프로그램은 돌려 받은 결과를 사용자 접속 처리와 더불어 웹 클라이언트에게 전달한다.

데이터베이스 브로커는 CORBA를 사용하는 어떠한 응용 프로그램에도 데이터베이스 서버로서 사용되는 장점이 있다. VIOLA 도구의 질의 생성기는 Orbix[14]를 통하여 데이터베이스 브로커에서 SQL 생성에 필요한 정보를 가져온다. 따라서 데이터베이스 브로커는 실행 환경에서는 Java CGI 프로그램과 연동되며, 프로그래밍 과정에서는 C++로 작성된 질의 생성기에서 사용된다. CGI 프로그램과 질의 생성기는 데이터베이스 브로커의 IDL 인터페이스를 준수함으로써 서로 다른 언어와 응용에서 데이터베이스 브로커의 서비스를 제공 받는다. 따라서 하나의 데이터베이스 서버가 범용으로 사용되고 있다.

3.2 코드 생성

VIOLA는 시각 프로그램을 해석하여 HTML 문서와 CGI 프로그램을 자동으로 생성하게 된다. 코드의 생성은 섹션 단위로 이루어지며 아이콘 분석, HTML 코드 생성, CGI 코드 생성의 세 단계를 거치게 된다.

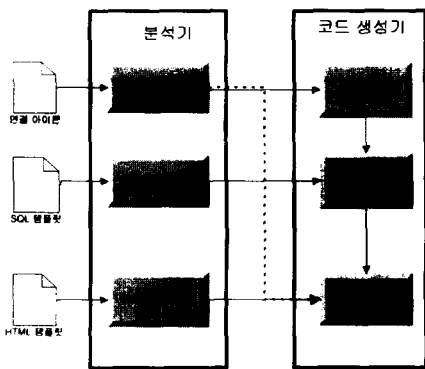
아이콘 분석은 자료 흐름이나 제어 흐름으로 연결되어 있는 아이콘의 속성 값을 자료 흐름이나 제어 흐름이 가지는 의미에 맞게 재설정하는 과정이다. 아이콘의 속성 값은 초기값으로 설정되고, 속성 편집이나 흐름 아이콘의 연결에 의해 값이 변화하게 된다. 속성 편집기에 의해 정적으로 변경되는 값은 HTML에 해당되는 특정 태그 정보를 변경하는 것으로 변경 즉시 아이콘의 속성 및 모양을 변경시킬 수 있다. 그러나 흐름 아이콘의 연결에 의한 속성의 변경은 끝 아이콘의 속성 값이 시작 아이콘의 속성 값에 종속되게 된다. 이때 시작 아이콘의 속성 값을 흐름을 연결하는 시점에서는 알 수 없고 프로그램이 실행되어야만 알 수 있는 경우도 있다. 그러나 끝 아이콘의 값은 시작 아이콘의 특정 값을 참조하고 있다는 정보를 가지고 있어야 한다. 이는 코드 생성시 두 아이콘의 참조 관계를 알고서 실행 시간에 올바른 결과 값을 가지도록 프로그램이 생성되어 지게 해야 한다. 예를 들어 질의 결과의 포매팅은 CGI 프로그램이 실행되어야만 결정될 수 있다. 동적인 값은 동적 변수로서 처리되어 종속적

인 참조 관계를 표현함으로써, CGI 프로그램 생성에 필요한 정보를 제공한다.

HTML 생성기는 HTML 골격 코드와 SQL 골격 코드를 생성한다. HTML 골격 코드는 GUI 아이콘에 대한 구조와 속성에 대한 정보를 가지고 해당되는 HTML 태그를 생성하고, 생성된 태그를 HTML 문서로 결합한다. 위치 정보를 반영하기 위하여 섹션 전체는 HTML 테이블로 처리되고 각 아이콘에 대한 HTML 태그는 테이블 데이터로 처리되어 테이블의 행과 열, rowspan 및 colspan을 계산하여 테이블에 결합된다. 테이블 태그에 의한 포매팅은 HTML 3부터 지원되며 위치위를 지원하는 가장 좋은 방법이다.

생성되는 HTML 골격 코드의 유형은 정적 문서가 될 HTML 골격 코드와 동적 문서가 될 HTML 골격 코드 두 가지로 분류될 수 있다. 정적 문서용 골격 코드는 HTML 문서가 되며, 동적 문서용 골격 코드는 데이터베이스 검색 결과에 따라 변하는 코드로서 CGI 프로그램 출력의 결과로서 생성되는 HTML 문서의 골격이 된다.

CGI 생성기는 프로세스 아이콘을 분석하여 프로세스 아이콘에 대한 코드 생성 루틴을 실행시키게 된다. 코드 생성에서는 질의 생성기에서 작성된 SQL 템플릿 코드, HTML 생성기에 의해 생성된 HTML 코드 및 HTML 골격 코드와 연결된 정보를 참조한다. 동적 HTML 생성을 위한 CGI 프로그램을 생성하는 과정은 (그림 5)와 같다.



(그림 5) CGI 프로그램 생성
(Fig. 5) CGI program generation

1단계는 사용자의 입력을 분석하고 초기 변수에 대한 처리를 담당하는 부분이고, 2단계는 SQL 템플릿 정보를 이용하여 완전한 SQL 코드를 생성하고 생성된

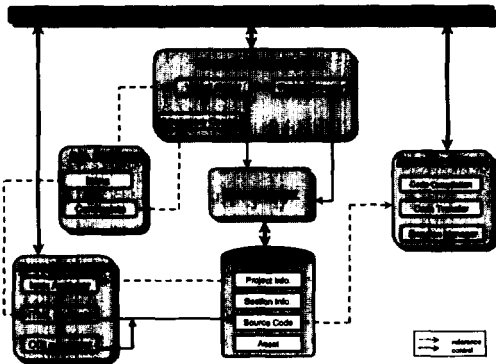
SQL 코드를 수행하는 부분이며, 3단계에서는 SQL 코드 수행 결과를 HTML 템플릿의 정보를 분석하여 문서로 포매팅한다. 1단계에서는 CGI 프로그램이 필요로 하는 매개 변수 값을 찾아내기 위해서 아이콘 분석 과정에서 추출한 정보가 이용된다. 예로 품의 구성 요소들에 대한 NAME과 VALUE에 대한 값을 알아내고 이를 조합하여 CGI 프로그램의 매개변수로 이용한다. 2단계에서는 로직에서 사용된 SQL질의문을 사용하여 이를 데이터베이스 브로커와 연결시키는 작업을 수행한다. 데이터베이스 브로커는 데이터베이스를 처리하는 기본 기능을 CORBA와 연동 가능한 형태의 API로 제공해 주기 때문에 CORBA를 초기화하고 이를 호출하는 루틴을 생성하게 된다. 3단계에서는 웹 인터페이스에 나타나는 요소에 맞게 질의 결과를 포매팅하게 된다. 프로그램 설계 단계에서는 동적인 자료를 처리할 수 없기 때문에 모든 정보는 정적으로 저장되어 있다. 이 정보를 동적으로 처리할 수 있는 형태로 변환하는 과정이다. VIOLA에서는 기본적으로 처리하는 형태의 포맷을 정의하여 두고 이를 변환하는 방법을 사용하였다.

코드 생성을 위하여 VIOLA에서는 사용자 입력 분석, 질의 생성, 질의 수행, 검색 결과 처리 및 포매팅 등의 주요 기능을 클래스로 구현하여 패키지로 제공한다. CGI 프로그램은 이들 클래스를 재사용하여 결합시킴으로써 생성되어진다. 코드 생성에 사용되는 패키지의 클래스는 다음과 같다.

- CgiInput : 폼에서 입력을 받아들여 필드와 값의 쌍으로 관리한다.
- FileReader : 지정된 파일 스트림을 읽고 스트링 형태로 변환한다.
- ResultParser : 데이터베이스 검색 결과를 파싱하고, 검색 영역에 해당하는 부분을 반환한다.
- Result2Hash : 데이터베이스 검색 결과를 각 필드 별로 해쉬 테이블에 저장한다.
- ListFrame : ListTable에 해당하는 정보 중에서 동적으로 반복되어 출력되어야 하는 정보를 저장한다.
- Parameter : URL의 인자 전달에 해당하는 부분을 생성하여 반환한다.
- Query : SQL 템플릿과 품의 사용자 입력을 결합하여 질의문을 생성하고, 질의문에 대한 수행 결과를 반환한다.

3.3 시스템 구조

VIOLA 도구는 시각 프로그래밍 편집기, 정보사전, 소스코드 생성기, 객체 관리기, 실행 관리기로 구성된다. 시각 프로그래밍 도구 VIOLA의 시스템 구조는 (그림 6)과 같다. 점선은 참조 관계를 나타내고 실선은 제어의 흐름을 나타낸다.



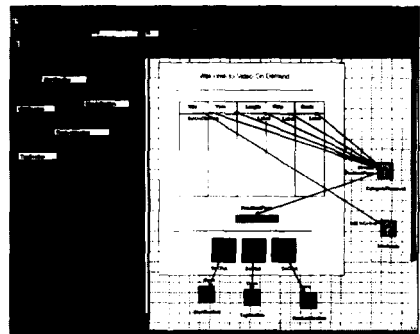
(그림 6) VIOLA시스템 구조도
(Fig. 6) System architecture of VIOLA

- 시각 프로그래밍 편집기는 사용자의 입력을 받아 시각 프로그램을 생성시켜 주는 시각 편집기, 아이콘의 배치와 다른 아이콘과의 연결에 대한 제약조건을 검사하는 일치성 검사기, 개발 중인 프로젝트의 전체적인 작업 흐름을 보여주는 프로젝트 브라우저로 구성되어 있다.
- 정보 사전은 아이콘과 제약조건에 대한 정보를 포함한다. 아이콘에 대한 정보로는 아이콘의 모양, 속성, 소스코드 생성 규칙을 포함한다. 제약조건은 아이콘 간의 연결시에 적용되는 연결관계 제약조건, 아이콘 간의 포함 관계를 정의한 포함관계 제약조건, 편집 단계의 의미를 지정하는 편집 제약조건으로 이루어져 있다. 시각 편집기는 정보사전에 있는 제약조건을 참조하여 사용자의 입력이 규칙에 위배되는지를 검사하며, 이를 사용자에게 알림으로써 올바른 구문으로 프로그래밍하도록 지원한다.
- 소스코드 생성기는 정보사전을 참조하여 아이콘의 의미를 분석함으로써 실행에 필요한 HTML 문서와 CGI 코드를 생성한다. 소스코드 생성기는 아이콘 분석기, HTML 문서 생성기, CGI 코드 생성기로 이루어진다.

- 객체 관리기는 시각편집기나 소스코드 생성기에 의해 생성된 정보를 관리한다. 관리하는 정보로는 프로젝트 정보, 섹션 정보, 섹션들 간의 관계 정보, 아이콘의 속성과 위치 정보, 생성된 소스 코드 등이 있다. 또한 시각 프로그래밍 편집기에서 사용된 이미지, 오디오, 비디오 파일 등 사용된 자원에 대한 관리도 이루어진다.
- 실행 관리기는 생성된 코드를 실행 환경에 맞게 설정하고 실행시키는 역할을 수행한다. 실행 관리기는 자바 소스 코드의 컴파일 기능, 생성된 코드를 웹 서버로 전송하는 기능, 실행을 위한 환경 변수를 설정해 주는 기능과 생성된 응용 시스템을 웹 브라우저에서 실행시켜 주는 기능이 있다. 코드 전송을 위해서 생성된 코드들이 코드의 유형에 따라 분류되어 관리되며, 웹 서버의 해당하는 디렉토리에 코드들이 전송된다.

3.4 예제 프로그래밍

프로그램 개발은 아이콘의 배열과 연결에 의해 이루어진다. 프로그래머는 아이콘박스에서 필요로 하는 아이콘을 선택하여 섹션 윈도우에 이동시킨다. 각 과정에서 일치성 검사가 이루어지며, 제약조건을 만족하지 않는 입력은 제외된다. (그림 7)은 VIOLA에서 VOD(Video On Demand) 응용 프로그램을 개발하는 과정의 한 화면이다. VOD는 원격 데이터베이스 서버에서 품 선택에 의해 필요로 하는 정보를 검색하고 검색된 비디오를 클라이언트에서 재생시켜준다.

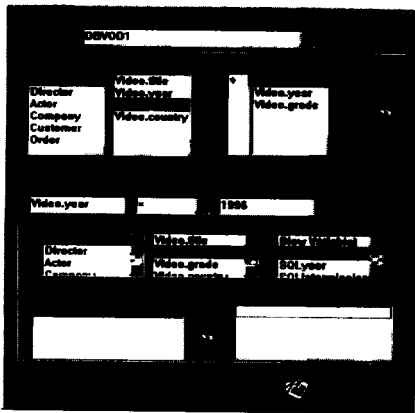


(그림 7) VIOLA 실행 화면
(Fig. 7) Program development in VIOLA

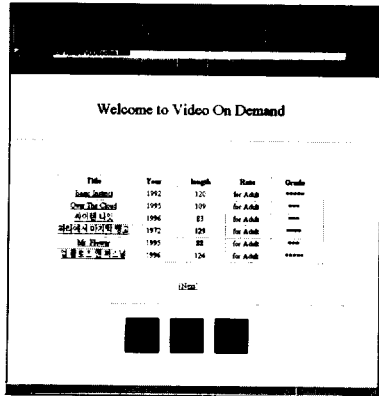
화면의 왼쪽 부분은 프로젝트 브라우저로서 섹션 간의 관계를 보여주며, 특정 섹션을 선택하게 되면 해

당되는 섹션으로 이동하게 된다. 오른쪽 부분은 섹션 윈도우로 실제로 프로그램이 이루어지는 부분이다. 최상위로 나타나는 VODSection은 폼 입력을 만족하는 영화 리스트를 포맷팅하는 것을 보여준다. VODSection의 시각형 부분에는 웹 사용자 접속 설계를 표현하고 시각형 영역 이외에는 프로그램 논리를 표현한다. ListTable 아이콘과 PreNextPage 아이콘은 질의 결과의 동적인 포맷팅을 위해 사용되며, 새개의 이미지는 다른 섹션과의 링크 흐름을 표시한다.

(그림 7)의 예에서는 CategoryRetrieval과 MovieInfo 두 개의 질의 아이콘이 사용되고 있다. CategoryRetrieval은 폼에서 입력 내용을 만족하는 영화를 나열하여 보여준다. MovieInfo는 여러 영화의 리스트에서 특정 영화에 대한 상세 정보를 얻는데 사용되는데, 실제 사용자 접속 화면은 다른 섹션에서 나타나고 여기서는 CGI에 대한 입력 정보만을 획득하는 수단이 된다. 질의 아이콘은 자체 내에서 질의 생성기를 수행시킬 수 있으며, (그림 8)은 질의 생성기의 한 화면이다. 질의 생성기는 데이터베이스의 필드 선택 부분과 where절을 생성하는 부분으로 이루어지며, 질의 과정 역시 시각적 인터페이스를 통해 최소한의 상호작용 만으로 쉽게 질의할 수 있도록 한다. (그림 9)는 (그림 7)의 VODSection이 웹 브라우저에서 실행되었을 때의 화면을 나타낸다. (그림 7)의 ListTable은 실제로 동적 자료의 검색 결과를 출력하는 테이블로 표현되고, PreNextPage 아이콘은 검색 페이지간 연결을 가능하도록 자동으로 하단에 생성되어진다.



(그림 8) 질의 생성기
(Fig. 8) Query builder dialog



(그림 9) 웹 브라우저 상의 실행 화면
(Fig. 9) Execution result by a web browser

4. 결 론

본 논문에서는 웹 응용 프로그램의 생성을 지원하는 VIOLA의 시각 언어 프레임워크에 대하여 설명하였다. VIOLA는 사용자 접속 설계와 프로그램 논리 설계를 자료흐름 방식을 이용하여 결합시켜 줌으로써 직관적인 인터페이스를 제공하며, 사용자가 복잡한 HTML 태그나 CGI 프로그래밍에 대한 지식이 없어도 쉽고 간단하게 웹 응용 프로그램을 생성할 수 있게 해준다. 사용자 접속 설계는 직접 조작에 의해 이루어지고 프로그램 논리 설계는 추상화된 아이콘과 자료흐름 방식을 사용한다. 페이지 단위로 보여지는 웹의 특성을 반영하고 쉽게 웹 응용 프로그램 설계를 지원하기 위하여 프로젝트-섹션-아이콘 메타포를 사용하였고, 섹션 간의 정보의 전달을 위해 수입 아이콘을 사용하였다.

VIOLA의 주요 특징은 다음과 같다.

- VIOLA는 단순한 HTML 문서의 저작 뿐만 아니라 미리 정의된 클래스를 재사용하여 결합함으로써 CGI 프로그램도 자동으로 생성할 수 있다. CGI 프로그램 생성을 위해 필요한 질의 생성 과정도 시각적으로 지원되며, 데이터베이스 검색을 위한 기능들이 데이터베이스 서버로 구현되어 있다.
- VIOLA는 사용자 접속 설계와 프로그램 논리 설계를 위하여 자료흐름 방식을 사용한다. 따라서 프로그래밍 과정이 직관적이고 사용하기 쉽다.
- VIOLA는 직접 조작과 위치됨을 지원한다. 사용자는 아이콘박스에서 필요로 하는 아이콘을 찾아 화

면상에 배열함으로써 HTML 문서를 작성하게 된다. 또한 아이콘들은 이동, 복사, 삭제가 자유롭다. 아이콘의 모양 뿐만 아니라 위치 정보까지도 저작의 결과와 실행의 결과가 동일하다. 이를 위하여 HTML 테이블 태그에 의한 문서 포맷팅을 사용하였다.

- VIOLA는 코드의 자동생성, 생성된 코드의 컴파일, 코드의 서버로의 이동 과정 등 웹 응용 프로그램 생성에 필요한 전과정을 지원한다.
- 생성된 CGI 코드는 자바로 이루어지기 때문에 HTML 문서와 더불어 플랫폼에 독립적으로 실행될 수 있다.

VIOLA는 MS Windows 95에서 Visual C++ 4.2로 구현되었으며, 그 활용성을 보여주기 위해 VOD 응용 프로그램을 생성하는데 적용하였다. 데이터베이스 검색 부분과 비디오 재생 함수는 각각 생성된 CGI 프로그램과 Helper 프로그램에 의해 이루어졌다. VIOLA는 웹을 기반으로 하는 다른 종류의 멀티미디어 응용 프로그램 개발에도 적용이 가능하다. 앞으로의 연구 방향으로 데이터베이스 검색 기능과 더불어 데이터의 삽입, 삭제, 수정 기능을 지원하도록 질의 생성기를 확장하는 것이 필요하다. 또한 스크립트를 웹 문서에 기술할 수 있게 함으로써 동적 HTML 문서 생성을 지원하도록 하고 XML(Extensible Markup Language)를 지원하려고 한다.

참 고 문 헌

[1] J. December and M. Grinsburg, 'CGI & HTML', Sams.net Publishing, 1995.

[2] A. Richmond and L. Richmond, "The WDWL : HTML_Editors," http://www.stars.com/VLib/Authoring/HTML_Editors.html.

[3] Gill, B., "A Tiered Model for the Web Authoring Tool Market," GartnerGroup, Jan. 1997.

[4] Gill, B., "Web Authoring Tool Tiers Defined, Part 1 & 2," GartnerGroup, Jan. 1997.

[5] Neil W. Dutton, 'OVUM Evaluate : Web Development Tools', OVUM Ltd., 1997.

[6] NetDynamics, Inc., "NetDynamics : Enterprise Network Application Platform," <http://www.netdynamics.com/>.

[7] BlueStone Software, "Sapphire/Web Home Page," <http://www.bluestone.com/products/sapphire/>.

[8] Daniel D. Hils, "Visual Languages and Computing Survey: Data Flow Visual Programming Languages," Journal of Visual Languages and Computing, May 1992, pp.69-101.

[9] Margaret Burnett, Adele Goldberg, and Ted Lewis, 'Visual Object-Oriented Programming, Concept and Environment', Manning Publications Co., 1995.

[10] M. Hirakawa, M. Tanaka, and T. Ichikawa, "An Iconic Programming System, HI-VISUAL," IEEE Transactions on Software Engineering, Vol.16, No.10, Oct. 1990, pp.1178-1184.

[11] IBM, "VisualAge for Smalltalk: Web Connection Feature," <http://www.software.com/software/advisage/vawebwp.html>.

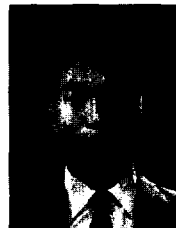
[12] J. Rumbaugh, et al., 'Object-Oriented Modeling and Design', Prentice-Hall International, Inc., 1991.

[13] Rober Orfali, Dan Harkey, Jeri Edward, 'The Essential Distributed Object Survival Guide', John Wiley & Sons, Inc., 1996.

[14] 'Orbix 2.1 Programming Guide', IONA Technologies Ltd., November 1996.

[15] 'OrbixWeb 2.0 Programming Guide', IONA Technologies Ltd., November 1996.

[16] C. S. Cho, et al., "VIOLA: A Visual Programming Tool for the Development of Web Applications," Proceedings of HCI International '97, Vol.2, Aug. 1997, pp.787-790.



조 창 식

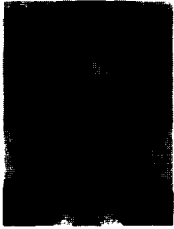
e-mail : cscho@etri.re.kr

1993년 경북대학교 전자계산학과 (학사)

1995년 경북대학교 전자계산학과 (석사)

1995년~현재 한국전자통신연구원 연구원

관심분야 : 멀티미디어 저장서버, 객체지향 기술, 웹 기술 등



신 규 상

e-mail : gsshin@etri.re.kr
1981년 성균관대학교 통계학과(학사)
1983년 서울대학교 계산통계학과(석사)
1983년~1996년 시스템공학연구소 연구원/선임연구원

1997년~1998년 시스템공학연구소 책임연구원
1998년~현재 한국전자통신연구원 책임연구원
관심분야 : 소프트웨어공학, 객체지향 기술, 멀티미디어 기술 등



마 평 수

e-mail : pmah@etri.re.kr
1985년 서울대학교 식물병리학과 졸업(학사)
1992년 City University of New York, USA 전산학과(석사)
1995년 Wright State University, USA 전산학과(박사)

1985년~1989년 시스템공학연구소 연구원
1989년~1990년 (주)태양금속 정보산업연구실 대리
1996년~현재 한국전자통신연구원 컴퓨터소프트웨어기술연구소 책임연구원
관심분야 : 멀티미디어 저장서버, 멀티미디어 검색, 소프트웨어 재공학 등