

WebAppGen: 웹 응용 생성기

심은지^o, 장희원, 음두헌
 덕성여자대학교 컴퓨터공학부
feelthelove@duksung.ac.kr, bbabba8686@nate.com, dheum@duksung.ac.kr

WebAppGen : Web Application Generator

Eun-Ji Shim^o, Hee-Won Jang, Doo-Hun Eum
 School of Computer Engineering, Duksung Women's University

요 약

전세계 인터넷 사용자 수가 약 12억 6천 만명에 이르며 2012년에는 17억명 이상이 인터넷을 사용할 것이라고 추정되는 가운데 웹 데이터베이스 응용에 대한 수요가 비즈니스를 포함하는 모든 분야에서 급속히 증가되고 있다.[1] 그러나 급증하는 수요에 비해 웹 응용의 생성 및 유지·보수는 대부분 수작업에 의존하기 때문에 많은 시간과 노력이 소요되고 있다. 본 논문에서 소개하는 WebAppGen은 ScriptGen과 InterfaceGen으로 구성된 웹 응용 자동 생성기로서 웹 응용을 전자동으로 생성해 생산성을 향상시킨다. ScriptGen은 웹 응용에 필요한 폼들과 이 폼들을 통해 이루어지는 질의에 대해 데이터베이스와 연동하여 처리하는 웹 스크립트들을 자동 생성하고 InterfaceGen은 ScriptGen을 통해 생성된 폼들을 엮어 접근하기 쉬운 트리 형태의 인터페이스를 제공한다. 상용 웹 응용 생성기들과 달리, WebAppGen은 웹 응용의 전 과정을 자동화 시킴으로써 생산성을 한 단계 더 향상시키고, 생성된 응용은 직관적인 계층구조를 가지는 트리 형태의 인터페이스를 지원한다. WebAppGen이 생성한 응용에서, 한 개의 폼은 관심의 대상인 엔티티 외에 이 엔티티와 직·간접적으로 연관된 모든 엔티티들에 대한 정보를 표현한다. 또한, WebAppGen은 생성될 응용의 기본 틀인 템플릿을 기반으로 하기 때문에 생성된 응용 및 WebAppGen 자체의 유지·보수를 용이하게 한다.

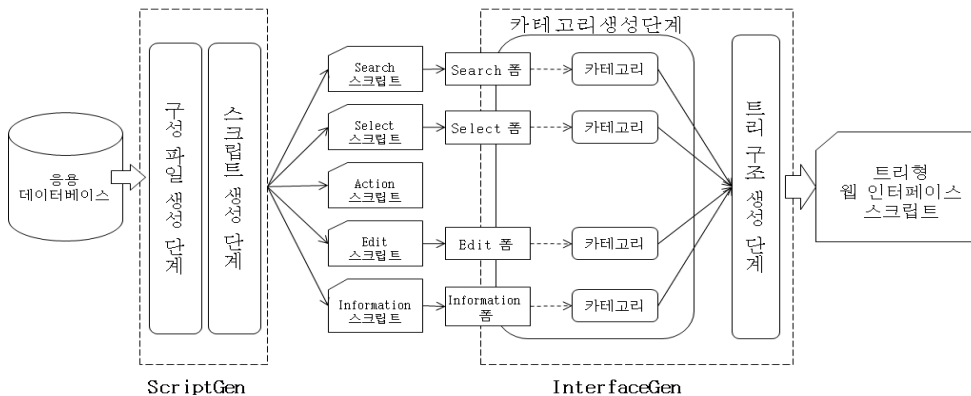
1. 서론

인터넷을 사용하는 사람들의 수가 기하급수적으로 늘어 전세계 인터넷 사용자는 약 12억 6천 만명으로 추정되며 (2008년 기준) 2012년에는 17억명 이상이 인터넷을 사용할 것이라고 한다.[1] 그러나 급증하는 수요에 비해 웹 응용의 생성 및 유지·보수는 대부분 수작업에 의존하기 때문에 많은 시간과 노력이 소요되고 있다. 본 논문에서 소개하는 WebAppGen은 웹 응용 자동 생성기로서 웹 응용의 생산성 향상을 지원한다. WebAppGen은 그림 1과 같이 사용자 인터페이스로 사용되는 기초 폼과 이 폼을 통한 사용자의 질의를 처리하는 스크립트들을 생성하는 ScriptGen과 생성된 폼들을 엮어 접근하기 쉬운 트리 형태의 인터페이스를 생성하는 InterfaceGen으로 구성된다.

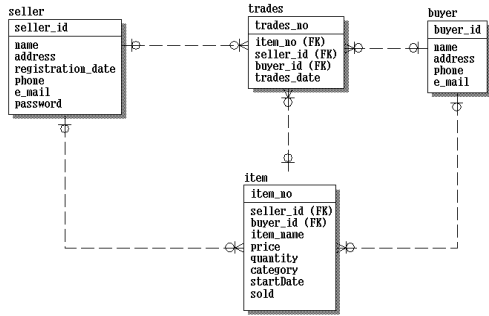
ScriptGen은 구성파일 생성 단계와 스크립트 생성 단계를 거쳐 응용 데이터베이스 내의 테이블 당 5개의 스크립트들(Search, Select, Action, Information, Edit)을 생성한다. 구성파일 생성 단계는 테이블들의 관계와 속성들을 분석하여 분석 내용을 담은 구성파일을 생성한다. 스크립트 생성 단계에서는 생성될 스크립트들의 기본 틀인 내장된 템플릿을 기반으로 구성파일의 내용을 반영하여 5개의 스크립트

들과 4개의 폼들을 생성한다. InterfaceGen은 카테고리 생성 단계와 트리 구조 생성 단계를 거쳐 트리형 인터페이스를 제공한다. 카테고리 생성 단계는 ScriptGen을 통해 생성된 4개의 폼들(Search, Select, Information, Edit)을 개발자가 지정한 카테고리와 연관시키며 트리 구조 생성 단계는 이렇게 생성된 카테고리들을 지정된 형태의 트리 구조로 구성한다.

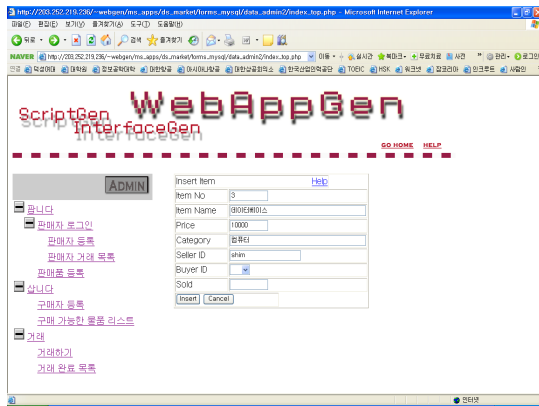
WebAppGen의 기능을 설명하기 위해 그림 2의 ER 다이어그램을 갖는 덕성장터 응용의 자동 생성 과정을 설명한다. item 엔티티는 덕성장터에서 거래 대상인 물품을 나타내고 seller와 buyer는 각각 판매자와 구매자를 표현한다. trades 엔티티는 판매자와 구매자간의 물품 거래를 나타낸다. 그림 2에서 item을 닷 엔티티(anchor entity)로 설정하기로 한다. 본 논문에서 닷 엔티티란 ER 다이어그램에서 기준이 되는 엔티티로 생성될 각 스크립트는 닷 엔티티를 중심으로 생성된다. item 엔티티는 seller, buyer 엔티티와 다대일의 관계를 갖고 trades 엔티티와는 일대다의 관계를 갖는다.



(그림 1) WebAppGen의 구성



(그림 2) 덕성장터의 ER 다이어그램



(그림 3) 덕성장터 응용의 실행화면

그림 3은 WebAppGen이 그림 2의 ER 다이어그램을 갖는 덕성장터 데이터베이스로부터 그림 1의 과정을 거쳐 생성한 덕성장터 응용의 실행 화면이다. ScriptGen은 각 테이블당 5개의 스크립트들을 생성해 총 20개의 스크립트들을 생성하고 InterfaceGen은 개발자가 지정한 카테고리([팝니다], [삽니다], ...)의 구조대로 트리형 인터페이스를 화면 왼쪽에 제공한다. 각 카테고리를 클릭하면 연결된 폼을 화면 오른쪽에 제공한다. [팝니다] 카테고리는 판매를 할 때 사용하며 [판매자 로그인] 카테고리와 [판매물 등록] 카테고리를 지식 카테고리로 갖는다. [판매자 로그인] 카테고리는 다시 [판매자 등록] 카테고리와 판매자의 거래 현황을 표현하는 [판매자 거래목록] 카테고리를 지식 카테고리로 갖는다. [삽니다] 카테고리는 구매를 할 때 필요한 [구매자 등록] 카테고리와 [구매 가능한 물품 리스트] 카테고리를 지식 카테고리로 갖는다. [거래] 카테고리는 판매자와 구매자 사이에서 거래가 이루어졌을 때 등록해야 할 [거래 하기]와 거래가 성립된 물품의 목록인 [거래 완료 목록]을 지식 카테고리로 갖는다. 그림 3의 오른쪽은 [판매물 등록] 카테고리를 클릭하였을 때의 폼을 나타낸 것이다. 새로운 판매물을 등록하기 위하여 폼에서 요구하는 정보(Item_no, Item_name, Price, Category, Seller_id, Buyer_id, Sold)들을 입력한 후 insert 버튼을 누른다.

WebAppGen과 비교할만한 상용 웹 응용 생성기인 AppGini3.1[2], ASP.NET2.0[3], Iron Speed Designer5.2[4], Code Charge Studio4.0[5]은 데이터베이스의 연동과 스크립트 생성에 관한 부분에서는 뛰어나지만 인터페이스를 간단한 폼으로 표현한다. WebAppGen은 생성된 폼을 카테고리에 연결시켜 트리 형태의 인터페이스를 구성하기 때문에

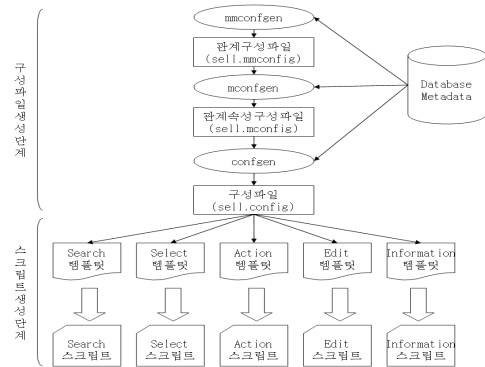
생성된 응용에 대한 정보를 직관적으로 파악해 사용할 수 있다. 또한, WebAppGen이 생성하는 응용의 폼은 상용 생성기들이 생성하는 응용의 폼에 비해 관심의 대상인 엔티티(맞 엔티티, anchor entity) 외에 맞 엔티티와 일대일, 다대일 관계의 엔티티들(첨부 엔티티, appended entity)을 포함하며 맞 엔티티와 일대다, 다대다 관계의 엔티티들(확장 엔티티, expanded entity)과 함께 확장 엔티티들과 일대일, 다대일 관계의 엔티티들(확장-첨부 엔티티, expanded-appended entity)을 제공한다. 그러나 상용 웹 응용 생성기들이 제공하는 폼들은 대부분 맞 엔티티만을 제공한다. WebAppGen은 생성될 응용의 기본 틀인 템플릿을 기반으로 하기 때문에 리플효과를 줄여 생성된 응용 및 WebAppGen 자체의 유지·보수와 버전관리를 용이하게 한다.

2절에서 WebAppGen의 개요를 설명하고 3절에서는 WebAppGen과 상용 웹 응용 생성기들의 기능을 비교한다. 마지막 절에서는 본 논문의 결론을 정리한다.

2. WebAppGen의 개요

본 절에서는 WebAppGen의 개요에 대해 설명한다. WebAppGen은 ScriptGen과 InterfaceGen으로 구성된다. ScriptGen은 웹 폼과 이를 통한 질의를 처리하는 스크립트들을 생성하고 InterfaceGen은 ScriptGen이 생성한 기초 폼들을 엮어 트리 형태의 인터페이스를 제공한다. 2.1 절에서는 기초 폼과 스크립트의 자동 생성 과정에 대해 설명하고 2.2절에서는 트리형 사용자 인터페이스의 자동 생성에 대해 설명한다.

2.1 웹 폼과 스크립트의 자동 생성



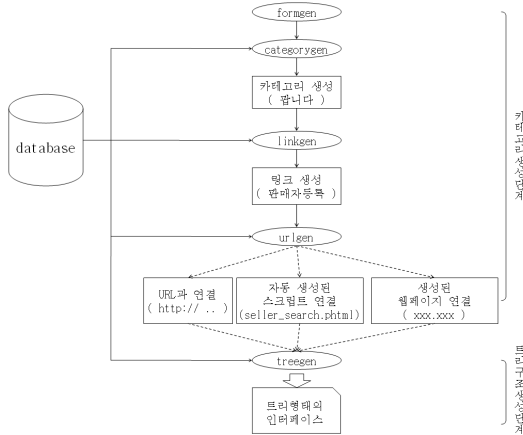
(그림 4) ScriptGen의 폼과 스크립트 생성 과정

그림 4는 ScriptGen을 이용해 덕성장터 응용의 폼과 스크립트들을 생성하는 과정을 나타낸 것이다. 그림 2의 덕성장터 ER 다이어그램에서 item 엔티티를 맞 엔티티로 설정한 경우, 생성 과정은 덕성장터 데이터베이스 내의 테이블당 구성파일 생성 단계와 스크립트 생성 단계의 두 단계를 거친다. 구성파일 생성 단계는 테이블들의 관계와 속성들을 분석하여 구성파일을 만든다. mmconfgen은 관계구성파일 템플릿에 응용 데이터베이스의 테이블 관계를 반영한 것으로 관계구성파일(sell.mmconfgen)을 생성한다. 템플릿은 생성할 파일의 기본이 되는 틀을 제공한다. mconfgen은 생성된 관계구성파일을 관계속성구성파일 템플릿에 반영하여 테이블들의 관계에 대한 정보와 간단한 속성들을 담은 관계속성구성파일(sell.mconfgen)을 생성한다. confgen은 관계속성구성파일을 구성파일 템플릿에 반영하여 세부적인 속성들을 담은 구성파일(sell.config)을 생성한다.

스크립트 생성 단계에서는 내장된 스크립트 템플릿을 기반으로 구성파일의 내용을 반영하여 5개의 스크립트

(Search, Select, Action, Information, Edit)들을 생성한다. 생성된 웹 스크립트들은 4개의 폼(Search, Select, Information, Edit)들을 제공하고 이 폼들을 통한 데이터의 검색, 삭제, 삽입, 갱신에 대한 질의를 수행하는 코드를 제공한다. 덕성장터 응용의 경우, Search 스크립트는 사용자가 입력한 Search 파라미터를 Select 스크립트에 전달한다. Select 스크립트는 Search 파라미터에 만족하는 레코드들을 사용자와 물품의 목록이 저장된 데이터베이스에서 검색하여 Select 폼과 함께 사용자에게 제공한다. Edit 스크립트에서는 데이터의 편집 기능을 제공한다. Information 스크립트는 선택된 레코드와 일대일, 다대일, 일대다, 다대다 관계에 있는 모든 레코드들을 한 개의 폼에 제공한다. Action 스크립트는 데이터의 삽입, 삭제, 수정 작업을 실제 수행한다. Action 스크립트를 제외한 나머지 스크립트들은 기초 폼을 제공하여 InterfaceGen이 이들을 엮어 트리형 인터페이스를 생성할 수 있게 한다. 따라서 4개의 테이블들로 구성되는 덕성장터 데이터베이스에서 테이블명만 주면 ScriptGen에 의해 테이블 당 5개의 스크립트들을 자동 생성하여 총 20개의 스크립트들을 생성한다. WebAppGen의 스크립트 생성 과정은 진자동으로 이루어져 웹 응용의 생산성 향상을 제공하며 템플릿을 기반으로 하기 때문에 데이터베이스의 구조적인 변경이나 스크립트들의 개별적인 변경에 대해 템플릿 파일의 내용만 수정하면 스크립트들을 신속하게 재생산 할 수 있어 WebAppGen 자체와 생성된 응용의 버전 관리가 용이하다.

2.2 사용자 인터페이스의 자동 생성



(그림 5) InterfaceGen의 카테고리 생성 과정

그림 5는 트리 구조의 인터페이스 생성 과정을 나타낸 것이다. InterfaceGen은 ScriptGen에서 생성된 스크립트, url, 임의로 만든 웹 페이지 등을 지정된 카테고리에 링크시켜 트리 형태의 인터페이스를 자동 생성한다.

생성 과정은 카테고리 생성 단계와 트리 구조 생성 단계로 구성된다. 카테고리 생성 단계는 개발자가 카테고리 생성 폼에 입력한 값을 받아 웹 인터페이스에서 사용 가능한 카테고리를 생성한다. formgen은 카테고리 입력 폼을 생성해 카테고리 이름, 링크 이름, 연결 주소 등을 입력 받는다. categorygen은 formgen에서 입력 받은 카테고리를 가지고 카테고리를 생성한다. linkgen은 formgen에서 입력 받은 링크 이름을 가지고 링크 이름을 생성한다. 카테고리 이름은 트리 형태에서 크게 항목을 구분하기 위해 사용되고 링크 이름은 화면에 직접 출력될 카테고리의 이름이다. 카테고리명과 링크는 데이터베이스의 category 테이블에 고유 식별번호(category_id)를 갖고 저장된다.

urlgen은 입력 받은 주소를 앞에서 생성한 카테고리에 연결한다. 입력 받은 연결 주소는 데이터베이스의 body 테이블에 저장되고 category 테이블의 기본키를 외래키로 갖게 된다. urlgen은 ScriptGen에서 자동 생성한 스크립트들을 연결할 뿐만 아니라, url이나 필요에 의해 새로 생성한 웹 페이지도 연결할 수 있어 ScriptGen에서 생성한 데이터의 검색, 삭제, 삽입, 갱신에 대한 스크립트들과 더불어 필요한 기능을 추가할 수 있게 한다.

트리 구조 생성 단계는 생성된 카테고리를 트리 형태로 구성하여 웹 페이지에 출력하는 기능을 제공한다. 개발자는 카테고리를 생성할 때 부모 카테고리, 자식 카테고리를 자유롭게 생성할 수 있다. treegen은 카테고리 생성 단계에서 category 테이블의 parent와 children 속성에 따라 부모인지 자식인지를 구분하여 트리 형태의 최종 인터페이스를 생성한다. InterfaceGen으로 생성한 트리 형태의 인터페이스는 직관적인 계층구조를 제공한다. 또한 개발자가 어떠한 트리 모양으로 카테고리를 생성하느냐에 따라 각각 다른 트리 형태를 표현할 수 있기 때문에 같은 기능을 하는 인터페이스일지라도 다양한 트리 형태의 인터페이스로 표현할 수 있다.

3. 상용 생성기와의 기능 비교

본 절에서는 WebAppGen과 상용 웹 응용 생성기들의 기능을 비교한다. 표 1은 WebAppGen과 상용 웹 응용 생성 도구인 AppGini3.1[2], ASP.NET2.0[3], Iron Speed Designer5.2[4], Code Charge Studio.4.0[5]의 기능을 비교 분석한 것이다.

표 1에서 보듯이, WebAppGen이 생성하는 폼들 중 Information 폼은 상용 스크립트 생성기들에 비해 관심의 대상인 닷 엔터티는 물론, 닷 엔터티 타입과 일대일, 다대일 관계 타입으로 연관되는 첨부 엔터티 타입의 엔터티들을 표시할 수 있고 닷 엔터티 타입과 일대다, 다대다 관계 타입으로 연관되는 확장 엔터티 타입의 엔터티들을 표시하며 이 확장 엔터티 타입과 첨부 관계에 있는 확장-첨부 엔터티 타입의 엔터티들도 한 폼에 표시할 수 있다. 이는 다른 스크립트 생성기 대부분이 단순히 닷 엔터티 타입의 엔터티만을 표시할 수 있는 것과 비교 된다. 또한, WebAppGen은 템플릿 기반으로 설계되어 프로그램의 한 부분을 수정하면 그 파일이 다른 부분에 미치는 리플 효과를 줄일 수 있기 때문에 생성기 자체와 생성된 응용의 버전 관리가 우수하다. 그에 비해 상용 생성기들은 프로그램 수정 시 버전에 맞춰 소스를 수정해야 하는 등의 시간과 노력을 많이 들여야 한다. WebAppGen이 생성하는 트리형 인터페이스는 기초 폼들을 개발자가 지정한 카테고리 별로 계층적으로 엮어 제공하기 때문에 상용 생성기들이 제공하는 간단한 폼들에 비해 직관적이고 체계적으로 정보를 처리할 수 있게 한다.

AppGini3.1은 BigProf사에서 개발한 웹 데이터베이스 응용 개발을 지원하는 도구이다. 이 도구는 Windows 환경에서 동작하며 MySQL 데이터베이스 테이블을 정의하면 테이블과 연동하는 PHP 코드를 템플릿 기반으로 생성한다. 이러한 작업은 GUI환경을 통해 이루어지므로 개발자는 PHP나 SQL 정의문과 관련되어 직접적인 기술을 하지 않아도 코드를 생성할 수 있다. 또한, 정의한 테이블의 각 컬럼에 대해 데이터를 검색, 입력, 수정, 삭제 할 수 있는 폼이 함께 생성된다. AppGini3.1은 프로젝트 단위로 작업을 관리한다. 데이터베이스나 코드의 수정이 필요할 경우, 프로젝트 파일을 수정하면 되기 때문에 생성된 응용의 유지·보수 측면은 용이한 편이다. PHP와 HTML 응용을 생성하지만 AppGini3.1 자체는 VB(Visual Basic)로 구현되었기 때문에 리플 효과로 인해 생성기 자체의 버전 관리가 쉽지 않다.

<표 1> WebAppGen과 상용 웹 응용 생성기와의 기능 비교

생성기		WebAppGen	AppGini3.1	ASP.NET2.0	Iron Speed Designer5.2	Code Charge Studio4.0
자동화 항목	데이터베이스 연동 스크립트 생성	자동	자동	자동	자동	자동
	인터페이스 생성	폼들의 트리형	간단한 폼	간단한 폼	다양한 폼	다양한 폼
	폼에 표현 가능한 엔티티	닷, 첨부, 확장, 확장-첨부 엔티티	닷 엔티티	닷, 확장 엔티티	닷, 확장, 첨부 엔티티	닷 엔티티
생성기의 버전 관리		+++	++	++	++	+++
생성된 응용의 유지·보수		+++	+++	+++	+++	+++
생성 스크립트 언어		PHTML, HTML	PHP, HTML	.NET언어 (C#, VB.NET)	Visual basic .Net언어, C#	PHP, ASP, JSP, Perl, ColdFusion, ASP.NET
운영체제		Linux, Windows	Windows,	Windows	Windows	Windows
생산성 지원		+++	++	++	++	++

AppGini3.1은 간단한 폼 형태의 인터페이스를 제공할 뿐 이 폼들을 엮은 보다 고급의 인터페이스를 제공하지는 못한다. 또한, 이 도구는 데이터베이스를 손쉽게 구축하고 질의할 수 있는 응용을 생성 하지만 한 폼에 표현 가능한 엔티티는 닷 엔티티로 제한된다. 즉, 데이터베이스 테이블을 정의할 때 외래키로 일대일, 일대다, 다대일의 관계는 설정할 수 있지만 이들을 한 폼에 표현하지는 못한다.

ASP.NET2.0은 Microsoft사의 동적 웹 응용 프로그램을 작성하기 위한 기술로서 .NET 프레임워크의 일부다. VB.NET, C#, Jscript.NET 등을 포함한 모든 .NET 호환 언어에서 ASP.NET2.0 응용 프로그램을 제작할 수 있다. ASP.NET2.0은 Visual Studio.NET과 SQL Sever 데이터베이스를 이용하여 웹 사이트를 자동으로 생성하며 프로젝트 파일 단위로 작업을 관리하기 때문에 프로젝트 파일에 대한 수정을 통해 생산된 응용의 유지·보수를 쉽게 할 수 있다. 그러나 ASP.NET2.0은 폼에 필요한 GUI 요소들을 수동으로 선택, 구성하여 폼을 생성한다. 또한, 개발자질의 처리 코드 등을 추가해야 완전한 코드가 생성되기 때문에 생산성 지원은 다소 미흡하다. WebMatrix 도구를 이용하면 새로운 테이블의 생성 및 기존 데이터베이스의 접근이 가능하지만 한 폼에 표현 가능한 엔티티는 닷 엔티티와 첨부 엔티티로 제한적이고 이들을 한 폼에 표현하지는 못한다.

Iron Speed Designer5.2는 Iron Speed사에서 개발한 웹 응용 생성기로 .NET 언어와 C# 언어를 사용하여 프로그램 제작이 가능하다. Iron Speed Designer5.2는 다양한 디자인의 폼의 형태를 GUI 요소들을 통하여 제공해 개발자가 선택할 수 있게 한다. Iron Speed Designer5.2는 프로젝트 단위로 웹 응용을 생성 및 관리 하므로 생성된 응용의 유지·보수 측면에서 용이하다. Iron Speed Designer5.2는 table-in-table 방식을 사용하여 생성한 폼 안에 닷 엔티티 타입과 일대다 관계 타입으로 연관되는 첨부 엔티티 타입의 엔티티들과 다대일의 관계 타입으로 연관되는 확장 엔티티 타입의 엔티티들을 제공한다. 한 폼에서 다양한 엔티티 타입의 레코드들을 빠르게 볼 수 있다는 장점이 있으나 닷 엔티티와 직접적으로 연관이 있는 특정 엔티티를 폼에 나타내는 것이 아니라 관계 타입의 엔티티 리스트 전체를 제공할 뿐이다.

Code Charge Studio4.0은 YesSoftware에서 개발한 웹 응용 생성기이다. Code Charge Studio4.0은 PHP, ASP, JSP, ASP.NET 등의 다양한 언어를 템플릿 기반으로 하여 응용 프로그램을 생성한다. Code Charge Studio4.0은 Microsoft

SQL Sever, Oracle, MySQL, DB2, Microsoft Access 데이터베이스를 이용하여 웹 사이트를 자동으로 생성하고, Visual query builder 도구를 통해 데이터베이스에 접근하고 데이터를 처리할 질의문을 자동 생성한다. Code Charge Studio4.0은 여러 디자인의 레이아웃을 제공하여 다양한 디자인의 인터페이스를 생성할 수 있다. 프로젝트 단위로 작업을 관리하기 때문에 프로젝트 파일에 대한 수정을 통해 생산된 응용의 유지·보수를 쉽게 할 수 있다. 또한 템플릿 기반으로 설계되어 리플 효과를 줄일 수 있어 생성된 응용의 버전 관리가 우수하다. 그러나 한 폼에 표현 가능한 엔티티는 닷 엔티티로 제한된다. 즉, 데이터베이스 테이블을 정의할 때 외래키로 일대일, 일대다, 다대일의 관계는 설정할 수 있지만 이들을 한 폼에 표현하지는 못한다.

4. 결론

본 논문에서 소개하는 WebAppGen은 웹 응용을 자동 생성하는 도구로서 웹 응용의 생산성 향상을 지원한다. WebAppGen은 ScriptGen을 통해 사용자 인터페이스로 사용될 기초 폼들과 함께 이 폼을 통한 질의를 처리하는 스크립트들을 생성하고 InterfaceGen은 생성된 폼들을 개발자가 지정한 카테고리라 연계하여 트리 형태로 구성함으로써 좀 더 편리하고 직관적으로 정보를 처리할 수 있는 트리형 인터페이스를 제공한다. WebAppGen은 표 1 에서 보인 바와 같이 상용 웹 응용 생성기들 보다 다음의 기능들을 향상시킨다.

1. 폼의 표현 가능한 범위는 관심의 대상인 닷 엔티티 뿐만 아니라 닷 엔티티와 직· 간접적으로 연관된 확장, 첨부, 확장-첨부 엔티티들까지 포함하여 데이터베이스 커버리지가 높다.
2. 폼들을 개발자가 지정한 카테고리별로 엮어 계층적이고 사용하기 쉬운 트리형 고급 인터페이스를 생성한다.
3. 템플릿 기반으로 구현되고 응용을 생성하기 때문에 생성기 자체 및 생성된 응용의 버전 관리가 용이하다.

참고 문헌

[1] Morgan Stanley, Looking Around the Whole Online World, <http://www.emarketer.com/>, 2007
 [2] AppGini3.1, <http://www.bigprof.com>
 [3] ASP.NET2.0, <http://www.asp.net>
 [4] Iron Speed Designer5.2, <http://www.ironspeed.com>
 [5] CodeChargeStudio4.0, <http://www.yessoftware.com>, <http://www.codecharge.co.kr>