

WebGen: 템플릿 기반 웹 스크립트 생성기

음 두 힌[†]

요 약

데이터베이스와 연동하는 웹 응용에 대한 수요가 비즈니스를 포함하는 모든 분야에서 급속히 증가하고 있다. 그러나 급증하는 수요에 비해 웹 응용의 작성 및 유지·보수에 많은 시간과 노력이 소요되고 있다. 본 논문에서 소개하는 웹 스크립트 자동 생성기인 WebGen은 웹 응용에 필요한 폼들과 이 폼들을 통해 이루어지는 질의에 대해 데이터베이스와 연동하여 처리하는 웹 스크립트들을 자동 생성하는 소프트웨어 도구다. WebGen은 웹 응용 개발자가 작성하는 구성파일(configuration file)에 정의된 선언적인 내용을, 생성될 스크립트의 기본 원형인 내장된 템플릿(template)에 반영하여 5개의 웹 스크립트들(Search, Select, Edit, Information, Action)을 생성한다. Action 스크립트를 제외한 나머지 스크립트들은 사용자 인터페이스로 각각 해당되는 웹 폼을 생성한다. 따라서 WebGen은 웹 응용 작성을 위한 시간과 노력을 크게 줄여 웹 응용의 생산성을 향상시킨다. 상용 웹 스크립트 생성기들과 달리, WebGen은 상호 독립적인 템플릿들을 기반으로 하기 때문에 버전 관리가 용이하고 한 폼에 표현 가능한 정보도 관심의 대상인 엔티티 외에 이 엔티티와 직·간접적으로 연관된 모든 엔티티들을 포함한다.

키워드: 웹 스크립트 생성기, 웹 응용, 템플릿 기반, 웹 응용 생산성

WebGen: a Template-based Web Script Generator

Doohun Eum[†]

ABSTRACT

The demand for Web applications that run on databases has been rapidly increased in every area including business. Comparing to this growing demand, it still takes much time to write and maintain Web applications. In this paper, we introduce the Web script generator, WebGen that generates the Web forms as application interface and the Web scripts that process the queries by the forms operating with a database. WebGen generates five Web scripts (Search, Select, Edit, Information, and Action) from built-in templates that are the frames for those scripts by applying the declarative contents of a user-written configuration file. Each script except the Action script generates a corresponding form as user interface. Therefore, WebGen enhances Web application productivity by reducing the development time and effort for Web applications. Unlike the commercial Web script generators, WebGen supports easy version management because it is based on independent templates. Moreover, a WebGen-generated form includes not only the interested entity but also the entities that are related directly and indirectly with the interested entity.

Key Words : Web Script Generator, Web Applications, Template-based, Web Application Productivity

1. 서 론

시장조사기관인 DATAMONITOR에 의하면 2003년 응용 소프트웨어 시장 규모는 234억 2천 5백만 달러 규모였고 2008년에는 301억 5천 9백만 달러로 크게 신장할 것으로 예측되고 있다. 특히, 인터넷을 통해 사용자에게 정적인 정보를 제공하는데 그치지 않고 상호작용을 제공하는 대화식 웹 응용에 대한 수요가 급증하고 있다. 이러한 수요는 비즈니스 분야 뿐 만 아니라 거의 모든 분야에 걸쳐 폭넓고 다양하게 일어나고 있다[1].

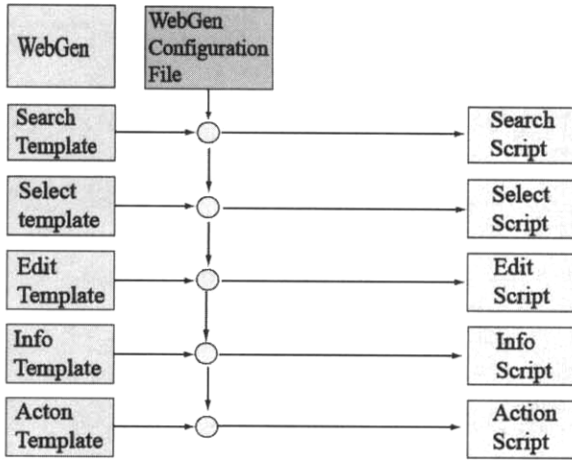
본 논문에서 소개하는 WebGen은 웹 스크립트 생성기로

서 응용에 필요한 웹 폼들과 이 폼들을 통해 이루어지는 질의를 데이터베이스와 연동하여 처리하는 웹 스크립트들을 자동 생성한다. WebGen은 (그림 1)과 같이 웹 응용 개발자가 작성하는 구성파일(configuration file)의 선언적인 내용을 미리 정의된 템플릿(template)에 반영하여 5개의 웹 스크립트들(Search, Select, Edit, Information, Action)을 생성하며 Action 스크립트를 제외한 나머지 스크립트들은 각각 해당되는 폼을 사용자 인터페이스로 클라이언트에게 제공한다. 템플릿은 생성될 폼과 스크립트를 위한 기본 틀이다. 개발자가 구성파일에 생성될 폼들이 어떤 항목들을 포함할 것인지를 PHP 배열에 선언적으로 정의하면 WebGen은 지정된 항목에 대한 정보를 검색하고 편집할 수 있는 폼과 PHP 스크립트들을 자동 생성한다. 따라서 WebGen은 웹 응용 작성을 위한 시간과 노력을 크게 줄여 웹 응용의 생산성을 크게

※ 본 논문은 2006년 덕성여대 자연과학연구소 연구비 지원으로 연구되었음.

† 정 회 원 : 덕성여자대학교 컴퓨터공학부 교수

논문접수: 2006년 11월 13일, 심사완료: 2007년 6월 5일



(그림 1) 템플릿 기반 WebGen

항상시킨다. 또한, WebGen은 템플릿을 기반으로 스크립트들을 생성하기 때문에 데이터베이스의 구조가 변경되어도 구성파일의 내용만 수정하여 응용을 다시 자동 생성하면 된다. 따라서 WebGen으로 자동 생성한 웹 응용의 유지·보수도 용이하며 WebGen의 템플릿들은 파라미터 교환 외에는 상호 독립적이기 때문에 생성기 자체의 버전 관리도 쉽다.

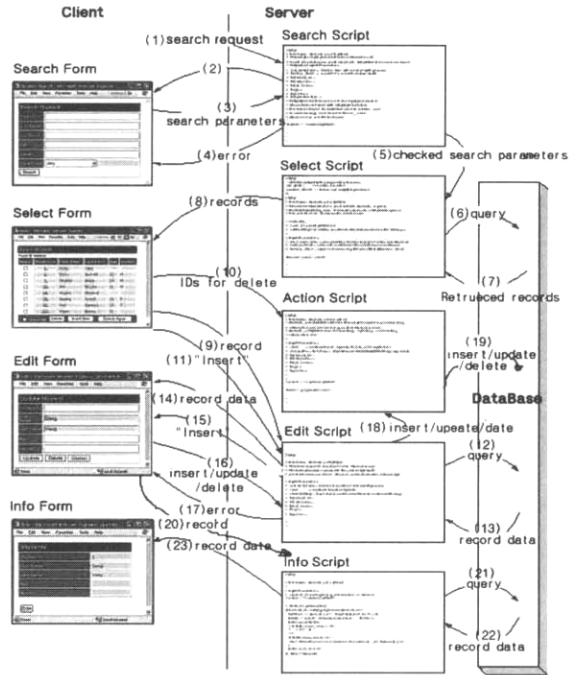
WebGen의 이전 버전인 WebSiteGen[2]은 클래스 다이어그램을 사용하여 폼들을 생성하고 데이터베이스와 폼들을 연동하는 자바 서블릿을 자동 생성한다. WebSiteGen의 다음 버전인 WebSiteGen2[3]는 데이터베이스 스키마를 이용하여 폼들과 JSP 코드 및 EJB 컴포넌트들을 자동 생성한다. 그러나 WebSiteGen과 WebSiteGen2는 모두 자바로 구현되었고 템플릿 기반이 아니기 때문에 생성기 자체와 생성된 응용의 유지·보수가 쉽지 않다.

템플릿 기반의 상용 웹 스크립트 생성기들로는 AppGini3.1[4], FW Plus[5], ASP.NET2.0[6], PHPMaker4.02[7] 등이 있다. 그러나 이들은 WebGen과 달리, 생성기의 한 부분을 수정하면 그 과급이 다른 부분에 미치는 리플 효과로 인해 버전 관리가 용이하지 못하다. 또한, WebGen은 한 폼에 관심의 대상인 엔티티(닷 엔티티, anchor entity) 외에 이 엔티티와 일대일, 다대일 관계의 엔티티들(첨부 엔티티, appended entity)을 포함하며 일대다, 다대다 관계의 엔티티들(확장 엔티티, expanded entity)과 함께 이 확장 엔티티들과 일대일, 다대일 관계의 엔티티들(확장-첨부 엔티티, expanded-appended entity)도 제공한다. 따라서 WebGen 응용의 폼들은 데이터베이스 테이블에 대한 커버리지가 매우 높은 반면, 상용 생성기들이 제공하는 폼들은 닷 엔티티와 함께 기껏해야 첨부 또는 확장 엔티티에 대한 정보만을 제공한다.

2절에서 WebGen의 개요를 설명하고 3절에서는 WebGen과 상용 스크립트 생성기들의 기능을 비교 분석한다. 4절에서는 WebGen의 구현을 간략히 설명하고 마지막 절에서는 본 논문의 결론을 정리한다.

2. WebGen의 개요

본 절에서는 WebGen의 개요를 설명한다. 웹 응용 개발자가

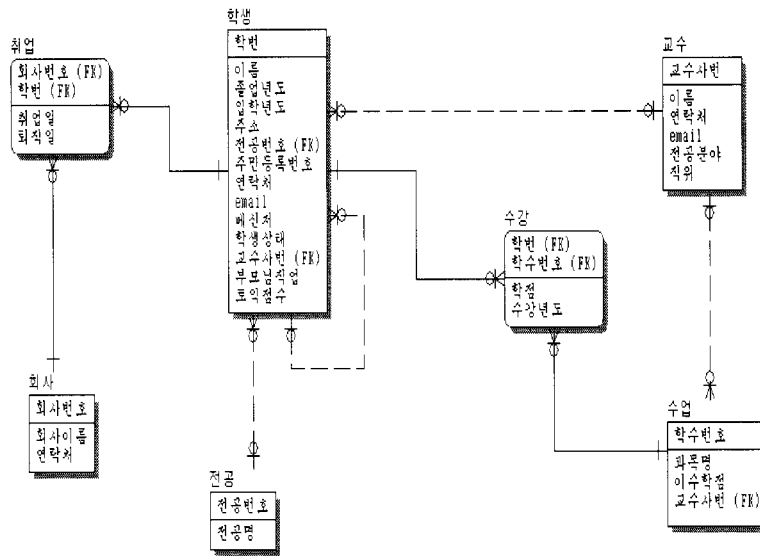


(그림 2) 생성된 웹 스크립트와 폼의 상호작용

구성파일에 사용자 인터페이스로 사용될 4개의 폼들(Search, Select, Edit, Information)이 어떤 항목들을 가질 것인지를 정의하면 WebGen은 내장된 템플릿들을 기반으로 구성파일의 내용을 반영하여 5개의 스크립트들(Search, Select, Edit, Information, Action)을 생성한다. 생성된 웹 스크립트들은 4개의 폼들을 생성하고 이 폼들을 통한 데이터의 검색, 삭제, 삽입, 갱신에 대한 질의를 수행하는 코드를 제공한다.

Search 스크립트는 사용자가 Search 폼에 입력한 검색 파라미터를 Select 스크립트에 전달한다. Select 스크립트는 검색 파라미터를 만족하는 레코드들을 데이터베이스에서 검색하여 Select 폼과 함께 사용자에게 제공한다. Edit 스크립트는 사용자가 선택한 레코드를 Edit 폼을 통해 수정할 수 있게 한다. Information 스크립트는 선택된 레코드(닷 엔티티) 외에 이 레코드와 일대일, 다대일 관계의 레코드들(첨부 엔티티), 일대다, 다대다 관계의 레코드들(확장 엔티티), 확장 엔티티와 일대일, 다대일 관계의 레코드들(확장-첨부 엔티티)을 한 개의 폼에 제공한다. Action 스크립트는 데이터의 삽입, 삭제, 수정 작업을 실제 수행한다.

(그림 2)는 생성된 폼들과 스크립트들 간의 상호작용을 보인 것이다. 사용자의 검색 요구(1)는 Search 스크립트에 전달되고 Search 스크립트는 Search 폼을 생성하여 사용자에게 제공한다(2). 사용자가 검색 파라미터를 입력하고 Search 버튼을 누르면 검색 파라미터에 오류가 있는지 검사하고 Select 스크립트로 이를 보낸다(3, 4, 5). Select 스크립트는 검색 파라미터를 만족하는 SELECT 질의를 실행하고(6), 결과 레코드들을 검색한다(7). 검색된 레코드들은 Select 폼을 통해 사용자에게 제공된다. 이때, 만약 시스템 모드가 갱신(Update)이면 검색된 레코드의 기본키(PK)를 Edit 스크립트



(그림 3) 정보공학대학 학생/졸업생 관리 응용의 ER 다이어그램

에게 보낸다(9). Delete 버튼을 누르면 Action 스크립트로 파라미터 값(cmd>Delete)이 넘어간다(10). 또한, Insert 버튼을 누르면 Edit 스크립트로 파라미터 값(cmd=Insert)이 넘어간다(11). Edit 스크립트는 파라미터 값(cmd=Update)이 Update 일 때 SELECT 질의를 실행하여(12), 결과 레코드들을 검색한다(13). 검색된 레코드들은 Edit 폼에 맞게 사용자에게 제공된다(14). Information 스크립트와 폼은 선택된 레코드와 함께 이 레코드와 직·간접으로 연관된 레코드들에 대한 자세한 정보를 제공한다. Action 스크립트는 삽입, 삭제, 수정 작업을 실제 수행한다. 파라미터 값(cmd)이 Insert일 때 Edit 스크립트는 삽입을 위한 폼을 생성하여 사용자에게 제공한다(15). Edit 폼에서 사용자가 Update 버튼을 누르면 Edit 스크립트로 파라미터 값(cmd=update)이 넘어간다(16). 이 파라미터에 오류가 있는지 검사하고(17), 없으면 Action 스크립트로 보낸다(18). Action 스크립트는 Edit 스크립트에서 받은 파라미터 값(cmd=update)을 만족하는 SELECT 질의를 수행한다(19). Information 버튼을 누르면(20) Information 스크립트는 레코드의 기본키(PK)를 가지고 SELECT 질의를 수행하고(21), 결과 레코드들을 Information 스크립트로 보낸다(23). 검색된 레코드들은 Information 폼에 맞게 사용자에게 제공된다.

(그림 3)은 WebGen으로 자동 생성한 덕성여자대학교 정보공학대학의 학생/졸업생 관리 응용의 ER 다이어그램이다. 엔티티들 간의 관계를 보인 이 ER 다이어그램에서 닷 엔티티란 웹 스크립트 생성의 기준이 되는 엔티티로서 생성될 각 폼과 스크립트는 닷 엔티티를 중심으로 생성된다. 학생 엔티티를 닷 엔티티로 설정할 경우, 학생 엔티티와 전공 엔티티는 다대일 관계를 가지고, 회사 및 수업 엔티티와는 다대다 관계, 교수 엔티티와는 다대일 관계를 가진다. 구성과일에는 닷 엔티티가 학생 엔티티임을 지정하고 각 폼의 구성을 명시하는 \$search_fields, \$select_fields, \$search_join_tables, \$edit_fields, \$info_fields의 PHP 배열 변수들을 정의한다.

\$search_join_tables는 닷 엔티티와 다대일, 다대다 관계에 있는 엔티티들을 조인시키기 위해 사용된다. 같은 방법으로, 교수, 수강, 취업 등의 엔티티들도 각각 닷 엔티티로 지정하여 웹 스크립트들을 생성한다. (그림 3)에서는 7개의 엔티티들에 대해 각각 5개의 웹 스크립트들이 생성되므로 총 35개의 스크립트들을 자동 생성하게 된다.

(그림 4)는 학생을 닷 엔티티로 지정했을 경우의 구성과일 내용 중, \$search_fields의 일부를 보인 것이다. 첫 번째 열의 컬럼명은 s_id(학번)이며 해당 테이블명은 student이다. Search 폼에서는 학번이라는 레이블이 붙여진다. s_id의 타입은 숫자이며 search 파라미터의 최대 크기(maxlen)는 50이고 텍스트 필드의 크기 또한 50임을 지정한다.

(그림 5)는 (그림 4)의 \$search_fields의 정의에 의해 WebGen이 생성한 Search 폼이다. 구성과일에 정의한 순서대로 폼의 필드들로 반영된다. 즉, 구성과일에 4개의 폼들이 어떤 필드들을 가질 것인지만 정의하면 해당 폼과 함께 각 필드들을

```

$search_fields=array(
  array("column"=>"s_id", "table"=>"student", "label"=>"학번",
    "type"=>"numeric", "maxlen"=>"50", "size"=>"50"),
  array("column"=>"s_name", "label"=>"이름",
    "type"=>"text", "maxlen"=>"50", "size"=>"50"),
  array("column"=>"s_year1", "label"=>"입학년도",
    "type"=>"text", "maxlen"=>"50", "size"=>"50"),
  array("column"=>"s_year2", "label"=>"졸업년도",
    "type"=>"text", "maxlen"=>"50", "size"=>"50"),
  array("column"=>"s_email", "label"=>"Email",
    "type"=>"email", "maxlen"=>"50", "size"=>"50"),
  array("column"=>"s_tel", "label"=>"전화번호",
    "type"=>"text", "maxlen"=>"50", "size"=>"50"),
  .....
  .....
  .....

```

(그림 4) \$search_fields의 일부

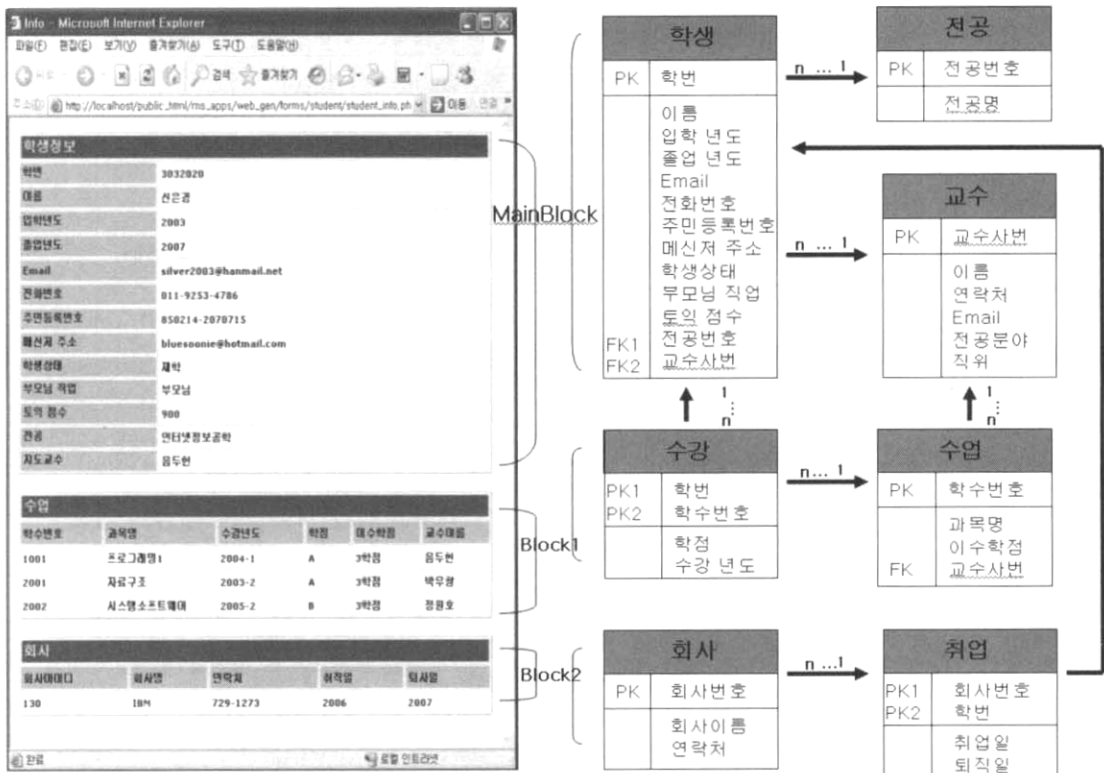


(그림 5) Search 폼

어떻게 데이터베이스와 연동 할지에 대한 코드가 WebGen 에 의해 자동 생성된다. Search 폼의 필드 중에서 하나 이상의 필드에 값을 입력하고 Search 버튼을 누르면 질의가 발생하여 조건에 맞는 학생들의 리스트가 Select 폼으로 제공된다. 필드에 값을 입력하지 않고 Search 버튼을 누르면 모든 학생들의 리스트가 제공된다.

(그림 6)의 Information 폼은 세 개의 블록으로 구성된다. 첫 번째 블록은 주 블록(MainBlock)으로서 닷 엔티티인 학생 자신과 함께 닷 엔티티와 다대일 관계인 전공 엔티티와 교수 엔티티(첨부 엔티티들)에 대한 정보를 제공한다. 두 번째 블록(Block1)에는 닷 엔티티와 일대다 관계인 수업 엔티티(확장 엔티티)에 대한 정보와 함께 이 수업 엔티티와 다대일 관계인 교수 엔티티(확장-첨부 엔티티)에 대한 정보가 제공된다. 즉, 학생이 이수한 수업에 대한 학점과 과목명이 리스트 되며 가장 오른쪽 컬럼에는 수업 엔티티와 다대일 관계에 있는 교수 엔티티의 정보까지 제공된다. Information 폼은 닷 엔티티와 간접적으로 연관된 확장-첨부 엔티티의 정보까지 제공함으로써 데이터베이스 테이블들에 대한 높은 커버리지를 지원한다. 세 번째 블록(Block2)은 닷 엔티티와 다대다 관계인 회사 엔티티(확장 엔티티)에 대한 정보를 제공한다. 회사는 학생이 졸업생일 경우 취업한 회사에 대한 정보다. ER 다이어그램에서 닷 엔티티와 다대다 또는 일대다 관계인 엔티티 당 한 개의 블록이 제공된다.

(그림 7)은(그림 6)의 Information 폼을 생성하기 위해 구성과일에 정의한 \$info_fields다. \$info_fields의 첫 번째 블록에는 닷 엔티티와 함께 닷 엔티티와 다대일 관계인 엔티티들에 관한 필드들을 정의한다. 즉, 닷 엔티티 정보인 s_id (학번), s_name(이름) 등을 정의하고 다대일의 관계인 전공 엔티티와 교수 엔티티는 joined_tables를 통해 조인 연산이 필요함을 명시한 후, 원하는 필드들(전공과 지도교수)을 정의한다. 두 번째와 세 번째 블록은 학생 엔티티와 각각 다



(그림 6) Information 폼

(그림 7) Information 폼

```

$info_blocks = array(
  array("title"=>"학생정보", "type"=>"main",
    "joined_tables"=>"student LEFT JOIN department on student.d_id=department.d_id LEFT JOIN professor on student.p_id=professor.p_id",
    "fields"=>array(
      array("column"=>"s_id", "table"=>"student", "label"=>"학번", "type"=>"numeric", "maxlen"=>"50", "size"=>"50"),
      array("column"=>"s_name", "label"=>"이름", "type"=>"text", "maxlen"=>"50", "size"=>"50"),
      array("column"=>"s_year1", "label"=>"입학년도", "type"=>"text", "maxlen"=>"50", "size"=>"50"),
      array("column"=>"s_year2", "label"=>"졸업년도", "type"=>"text", "maxlen"=>"50", "size"=>"50"),
      array("column"=>"s_email", "label"=>"Email", "type"=>"email", "maxlen"=>"50", "size"=>"50"),
      array("column"=>"s_tel", "label"=>"전화번호", "type"=>"text", "maxlen"=>"50", "size"=>"50"),
      array("column"=>"s_jumin", "label"=>"주민등록번호", "type"=>"text", "maxlen"=>"50", "size"=>"50"),
      array("column"=>"s_mess", "label"=>"메신저 주소", "type"=>"text", "maxlen"=>"50", "size"=>"50"),
      array("column"=>"s_abs", "label"=>"학생상태", "type"=>"text", "maxlen"=>"50", "size"=>"50"),
      array("column"=>"s_par", "label"=>"부모님 직업", "type"=>"text", "maxlen"=>"50", "size"=>"50"),
      array("column"=>"s_toeic", "label"=>"도입 점수", "type"=>"text", "maxlen"=>"50", "size"=>"50"),
      array("table"=>"department", "column"=>"d_name", "label"=>"전공", "type"=>"text", "size"=>"50"),
      array("table"=>"professor", "column"=>"p_name", "label"=>"지도교수", "type"=>"text", "size"=>"50"),
    ))),

  array("title"=>"수업", "type"=>"nested",
    "joined_tables"=>"student LEFT JOIN course_taken on student.s_id=course_taken.s_id
      LEFT JOIN course on course_taken.c_id=course.c_id LEFT JOIN professor on professor.p_id=course.p_id",
    "fields"=>array(
      array("column"=>"c_id", "label"=>"학수번호", "type"=>"numeric", "size"=>"50"),
      array("column"=>"c_name", "label"=>"과목명", "type"=>"text", "size"=>"50"),
      array("table"=>"course_taken", "column"=>"ct_year", "label"=>"수강년도", "type"=>"text", "size"=>"50"),
      array("table"=>"course_taken", "column"=>"grade", "label"=>"학점", "type"=>"text", "size"=>"50"),
      array("column"=>"credits", "label"=>"이수학점", "type"=>"text", "size"=>"50"),
      array("table"=>"professor", "column"=>"p_name", "label"=>"교수이름", "type"=>"text", "size"=>"50"),
    ))),

  array("title"=>"회사", "type"=>"nested",
    "joined_tables"=>"student LEFT JOIN work on student.s_id=work.s_id
      LEFT JOIN company on work.co_id=company.co_id",
    "fields"=>array(
      array("column"=>"co_id", "label"=>"회사(이)디", "type"=>"numeric", "size"=>"50"),
      array("column"=>"co_name", "label"=>"회사명", "type"=>"text", "size"=>"50"),
      array("column"=>"co_tel", "label"=>"연락처", "type"=>"text", "size"=>"50"),
      array("table"=>"work", "column"=>"w_start", "label"=>"취직일", "type"=>"data", "size"=>"50"),
      array("table"=>"work", "column"=>"w_end", "label"=>"퇴사일", "type"=>"data", "size"=>"50"),
    ))),
);

```

(그림 7) \$info_fields

대다 관계인 수업 엔티티와 회사 엔티티에 대한 선언이다. 두 번째 블록에서 수업 엔티티와 그림 7에서 선언한 배열들이 그림 6의 각 블록으로 매핑되는 것을 확인할 수 있다.

3. 상용 웹 스크립트 생성기와의 비교

본 절에서는 WebGen과 상용 웹 스크립트 생성기들의 기능을 비교한다. <표 1>은 WebGen과 상용 웹 스크립트 생성 도구인 AppGini3.1[4], FW Plus[5], ASP.NET2.0[6], PHPMaker4.02[7]의 기능을 비교 분석한 것이다.

<표 1>에서 보듯이, WebGen이 생성하는 Information 폼은 상용 스크립트 생성기들에 비해 관심의 대상인 닷 엔티티는 물론, 닷 엔티티 타입과 일대일, 다대일 관계 타입으로 연관되는 첨부 엔티티 타입의 엔티티들을 표시할 수 있고 닷 엔티티 타입과 일대다 관계 타입으로 연관되는 확장 엔티티 타입의 엔티티들을 표시하며 이 확장 엔티티 타입과 첨부 관계에 있는 확장-첨부 엔티티 타입의 엔티티들도 한

폼에 표시한다는 장점이 있다. 또한, WebGen은 템플릿 기반으로 설계되어 프로그램의 한 부분을 수정하면 그 파급이 다른 부분에 미치는 리플 효과를 줄일 수 있기 때문에 프로그램 자체의 버전 관리가 상용 생성기들에 비해 우수하다.

AppGini3.1은 BigProf사에서 개발한 웹 데이터베이스 응용 개발을 지원하는 도구다. 이 도구는 Windows 환경에서 MySQL 데이터베이스 테이블을 정의하면 테이블과 연동하는 PHP 코드를 템플릿 기반으로 생성한다. 이러한 작업은 GUI를 통해서 이루어지므로 사용자는 데이터베이스나 PHP, SQL 질의문과 관련된 직접적인 기술을 하지 않아도 코드를 생성할 수 있다. 또한, 정의한 테이블의 각 컬럼에 데이터를 검색, 입력, 수정, 삭제 할 수 있는 폼이 함께 생성된다.

AppGini3.1은 프로젝트 파일 단위로 작업을 관리한다. 데이터베이스나 코드의 수정이 필요할 경우, 프로젝트 파일을 수정하면 되기 때문에 생성된 응용의 유지·보수 측면에서 용이하다. PHP와 HTML 응용을 생성하지만 AppGini 자체는 VB(Visual Basic)로 구현되었기 때문에 리플 효과로 인해 버전 관리가 어렵고 언어적 특징 때문에 다른 운영체제

〈표 1〉 상용 웹 스크립트 생성기들과의 기능 비교

	WebGen	AppGini3.1	FW Plus	ASP.NET2.0	PHPMaker4.02
한 폼에 표현 가능한 엔티티	닷, 첨부, 확장, 확장-첨부 엔티티	닷 엔티티	닷, 첨부, 확장 엔티티	닷, 첨부 엔티티	닷/첨부 엔티티 닷/확장 엔티티
프로그램 자체의 유지·보수	+++	++	--	++	+++
생성된 스크립트 유지·보수	---	+++	+++	--	+++
스크립트 언어	PHP, HTML	PHP, HTML	Java, ASP, JSP PHP, C#	.NET언어 (C#, VB.NET)	PHP
생산성	---	+++	+++	++	+++
운영체제	Windows, Unix	Windows	Windows, Unix, Linux	Windows	Windows

에서는 실행될 수 없다는 단점이 있다. 또한, 이 도구는 데이터베이스를 손쉽게 구축하고 질의할 수 있는 응용을 생성하지만 한 폼에 표현 가능한 엔티티는 닷 엔티티로 제한된다. 데이터베이스 테이블을 정의할 때 외래키로 일대일, 일대다, 다대일의 관계는 설정할 수 있지만 이들을 한 폼에 표현하지는 못한다.

FW Plus는 BMSsoft사에서 개발한 템플릿 기반의 소스 코드 자동 생성기다. 이 도구는 Windows, Unix, Linux 환경 등에서 실행되며 Oracle, DB2, Informix, Sybase, SQL Server, MySQL 등의 데이터베이스와 연동하여 Java, ASP, JSP, C#, PHP 코드를 템플릿 기반으로 생성한다. 이 도구는 자동 생성 도구와 관리 도구를 제공하여 웹 개발 시 필요한 개발 과정을 단축해 주고 자동화를 지원하며, 이를 이용해 웹에서 직접 데이터베이스 테이블을 생성할 수 있다. 또한 생성한 테이블의 조회 및 권한 관리, 메뉴 관리 등을 할 수 있는 폼이 생성되며, 응용의 유지·보수도 자동 생성 도구와 관리 도구를 사용하기 때문에 용이하다. FW Plus는 다양한 템플릿을 제공하며 개발자가 템플릿을 등록하고 관리할 수 있게 하지만 도구 자체는 Java로 구현되었기 때문에 생성기의 한 부분을 수정하면 그 가격이 다른 부분에 미치는 리플 효과로 인해 버전 관리가 용이하지 못하다. 또한, 이 도구가 생성하는 폼에 표현 가능한 엔티티는 닷, 첨부, 확장 엔티티로써 WebGen 처럼 확장-첨부 엔티티까지는 표현하지 못한다.

ASP.NET2.0은 Microsoft사의 동적 웹 응용 프로그램을 작성하기 위한 기술로서 .NET 프레임워크의 일부다. Visual Basic .NET, C# 및 JScript .NET을 포함한 모든 .NET 호환 언어에서 ASP.NET2.0 응용 프로그램을 제작할 수 있다. ASP.NET2.0은 Visual Studio.Net과 SQL Server 데이터베이스를 이용하여 웹 사이트를 자동으로 생성한다. ASP.NET2.0도 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 선택, 구성하여 반자동으로 폼을 생성한다. 또한 사용자가 프로그래밍에 관한 직접적인 기술을 추가해야 완전한 코드가 생성되기 때문에 생산성지원은 다소 미흡하다. Web Matrix 등을 이용하면 새로운 테이블 생성 및 기존 데이터

베이스 접근이 가능하지만 한 폼에 표현 가능한 엔티티는 닷 엔티티와 첨부 엔티티로 제한적이다.

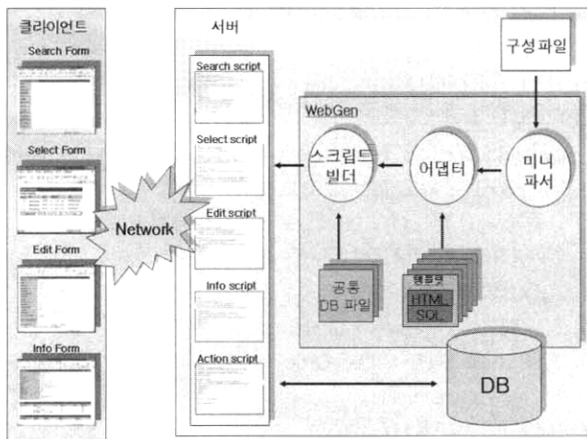
PHPMaker4.02는 e.World Technology사에서 개발한 웹 응용 자동 생성 도구다. 이 도구는 PHP 서버와 MySQL 데이터베이스를 기반으로 PHP 웹 스크립트를 자동 생성한다. 웹 응용 개발자가 MySQL 데이터베이스 테이블만 정의하면 이를 템플릿 파일에 반영하여 필요한 PHP 스크립트들을 생성한다. PHP나 데이터베이스에 익숙치 않은 초보자도 쉽고 빠르게 코드를 생산할 수 있다는 장점이 있다. 또한, 개발자가 취향에 따라 기본적인 웹 폼을 선택할 수 있다. PHPMaker4.02는 기본적으로 데이터베이스 테이블의 전체 필드 내용을 제공하는 폼과 데이터를 검색, 삽입, 삭제, 수정 할 수 있는 스크립트를 생성하며 실행환경은 Windows 운영체제로 국한된다. PHPMaker4.02는 작업을 프로젝트 파일로 저장하기 때문에 생성된 스크립트의 유지·보수 측면에서도 용이한 편이다. 저장된 프로젝트 파일을 수정한 후 템플릿에 반영하면 쉽게 스크립트를 재생산 할 수 있다. 또한, 템플릿 파일은 압축파일로 지정된 경로에 저장되어 개발자는 필요에 따라 템플릿 파일을 수정할 수 있기 때문에 도구 자체의 유지·보수도 수월한 편이다. PHPMaker4.02는 도구 자체에서 데이터베이스 테이블을 생성하고 기본키와 외래키 제약조건까지 설정할 수 있다. 그러나 닷 엔티티와 첨부 엔티티 또는 닷 엔티티와 확장 엔티티의 조합만을 한 폼에 제공하기 때문에 관련된 모든 정보를 한 폼에서 보며 조작할 수 없다.

4. WebGen의 구현

본 절에서는 WebGen의 구성과 PHP 웹 스크립트를 자동 생성하는 과정에 대해 설명한다. WebGen은 PHP로 구현되었으며 PostgreSQL 데이터베이스[8]와 PHP 서버를 사용한다. (그림 8)은 WebGen의 구성을 보인 것이다.

마니파서는 구성파일 문장들의 문법적인 구조를 분석 및 해석하는 역할을 한다. 문법에 어긋나거나 필요 파일이 해당 경로에 존재하지 않을 경우 에러 메시지를 출력한다.

어댑터는 구성파일의 내용을 스크립트의 원형인 템플릿에 반영하는 모듈이다. 각 템플릿은 HTML 폼 형성에 필요한 변



(그림 8) WebGen의 구성

수를 구성파일에 정의된 값으로 반영하는 부분과 SQL 데이터 베이스 질의문 생성에 필요한 컬럼명 및 테이블명으로 반영하는 부분으로 구성된다. 5개의 템플릿은 구성파일을 include 하여, 구성파일에 선언된 \$search_fields, \$select_fields, \$edit_fields, \$info_fields, \$search_join_tables 배열들을 사용하나 이 배열들 중 \$search_join_tables만은 select 템플릿에만 직접 포함되어 Select 스크립트를 생성하는데 사용된다.

어댑터는 각 템플릿의 반복문 내에서 구성파일에 정의된 배열에 포함되는 구성배열들을 한 개씩 읽어 들여 size, maxlen, type, label 필드값 등을 해당 변수(size, maxlen, type, label)에 저장한다. 저장된 변수는 스크립트 빌더에 의해 HTML 폼을 생성할 때 사용된다.

SQL 질의문 생성에 필요한 컬럼명과 테이블명도 반복문 내에서 구성파일에 정의된 컬럼명과 테이블명으로 대치된다. 즉, 매 반복마다 배열에서 읽어 온 하나의 컬럼값을 새로운 컬럼명에 할당하고 스크립트에 반영 시킨다.

스크립트 빌더는 어댑터 모듈에서 준비한 HTML 폼과 SQL 질의문 생성에 필요한 데이터들과 공통 DB 파일을 이용해서 실제 스크립트들을 생성하는 모듈이다. 각 템플릿에는 기본적인 HTML 폼과 SQL 질의문이 정의되어 있다. 공통 DB 파일에는 db_pgsql.inc와 common.phtml이 있다. db_pgsql.inc는 PHP 라이브러리가 제공하는 클래스 중에서 PostgreSQL 데이터베이스 연결과 호출에 사용되는 함수들이 저장되어 있는 파일이다. common.phtml에는 현재 서버에 연동되는 데이터베이스의 이름과 사용자ID, 패스워드, 호스트가 선언되어 있다. common.phtml은 db_pgsql.inc를 include 하며 생성된 5개의 스크립트들은 데이터베이스와 연동되기 위해 common.phtml을 include 한다. 또한, 스크립트 빌더 모듈은 php의 확장 문자 처리도 담당한다. php_escape.sed는 어댑터 모듈에서 넘겨받은 데이터 집합에 변화를 가해 미리 정의해 놓은 변환 집합으로 변환한다. 스크립트 빌더는 최종적으로 웹 문서 구성요소에 대해 정의된 스타일 시트를 적용해 HTML 폼을 생성한다. SQL 질의문은 Select, Update, Insert, Delete 문을 취사선택하여 각 스크립트에 맞게 생성된다.

(그림 9)는 Edit HTML 폼을 생성하기 위한 알고리즘의

```

<tr>
어댑터로부터 받은 label값을 출력.
switch(field type)
{
case 'text':
case 'numeric':
어댑터로부터 받은 size, maxlen, value 값으로 지정.
<td>HTML td 태그를 생성.</td>
case 'email':
case 'phone':
어댑터로부터 받은 value 값을 지정.
<td>HTML td 태그를 생성.</td>
case 'textarea':
어댑터로부터 받은 row, col, value 값을 지정.
<td>HTML td 태그를 생성.</td>
case 'to one':
어댑터로부터 받은 value 값을 지정.
<td>HTML td 태그를 생성.</td>
<button>View 버튼을 생성.</button>
if(맞 엔티티의 외래키이면) {
맞 엔티티와 다대일 관계에 있는
첨부 엔티티와 연결할 edit 스크립트를 영.
}else{
맞 엔티티와 다대일 관계에 있는
첨부 엔티티와 연결 안 되었다는 메시지를 출력.
}
<button>맞 엔티티와 다대일 관계에 있는 첨부 엔티티인
Search 스크립트와 연결하는 Select 버튼을 생성.</button>
case 'to many':
<td>HTML td 태그를 생성.</td>
<button>Show 버튼을 생성.</button>
if(맞 엔티티의 기본키이면){
확장 엔티티와 연결 안 되었다는 메시지를 출력.
}else{
확장 엔티티와 연결할 스크립트를 영.
}
}
case 'to info':
<td>HTML td 태그를 생성.</td>
<button>맞, 첨부, 확장엔티티를 모두 보여주는
information 스크립트를 연결하는 버튼을 생성.</button>
}
}
</tr>
    
```

(그림 9) Edit 폼 생성 알고리즘

일부를 보인 것이다. HTML 폼은 구성파일에 정의된 배열의 종류에 따라 다르게 생성된다. 템플릿에는 여러 가지 필드 타입들이 미리 정의되어 있다. 그 중에서 필드 타입이 email이면 어댑터로부터 받은 value값으로 td 태그를 생성하고 email 입력 형식을 보여준다. 필드 타입이 to_one이면 어댑터로부터 받은 value값으로 td태그를 생성하고 View, Select 버튼도 함께 생성한다. View 버튼은 맞 엔티티와 첨부 엔티티 관계인 스크립트 중 Edit 스크립트를 연결하고 Select 버튼은 Search 스크립트를 연결한다. 다른 HTML 폼을 생성하는 알고리즘도 이와 유사하다.

(그림 10)은 스크립트 빌더의 SQL 질의문을 생성하는 알고리즘 중, Select문을 생성하는 알고리즘을 보인 것이다. Select문의 기본구조는 Select절, From절, Where절의 3개의 절로 구성된다. 각각의 절에서 변수를 선언하고 변수 다음에 조건에 맞는 명령문을 추가하는 방식으로 Select문이 완성된다. Where절인 경우는 사용자가 조건 없이 검색할 때는 맞 엔티티에 있는 데이터를 모두 넘겨주므로 Where절에 조건이 추가되지 않는다. 사용자가 특정 조건으로 검색할 때는 Where절에 조건이 추가된다. Update문, Delete문, Insert문을 생성하는 알고리즘도 이와 유사하다.

5. 결론 및 향후연구

본 논문에서 소개한 WebGen은 웹 응용에 필요한 폼들과 함께 이 폼들을 통해 이루어지는 질의를 데이터베이스와 연

```

switch(template file)
{
  case search tmpl:
    Select문 생성.
  case select tmpl:
    Select 절을 받기 위한 변수명 선언($sSelect)
    For(select_fields의 배열 수){
      해당은 테이블명과 필름값을 문자열로 변환.
      변환된 문자열을 배열에 선언된 결합수만큼 추가.
    }
    "SELECT" 문자열과 변환된 문자열을 결합.
    From 절을 받기 위한 변수명 선언($sFrom).
    "FROM" 문자열과 및 엔티티명을 결합.
    Where 절을 받기 위한 변수명 선언($sWhere)
    for(search_fields의 배열 수){
      if {
        if($sWhere 값이 NULL이 아닐){
          $sWhere 변수에 문자열 "and" 추가.
        }
        필름명을 문자열로 변환하여 $sWhere에 결합.
      }
    }
    "WHERE" 문자열과 $sWhere변수를 결합.
    SQL 질의문을 수행하기 위한 변수명을 선언($sSQL).
    $sSQL에 $sSelect, $sFrom, $sWhere를 결합하여 할당.
    데이터베이스 질의 수행.
    :
    :
  case edit tmpl:
    Select문 생성.
  case info tmpl:
    Select문 생성.
  case action tmpl:
    Update문 생성.
    Delete문 생성.
    Insert문 생성.
}

```

(그림 10) SQL 질의문 생성 알고리즘

동하여 처리하는 웹 스크립트들을 자동 생성한다. WebGen은 웹 응용 개발자가 작성하는 구성파일에 정의된 선언적인 내용을 스크립트의 기본 원형인 내장된 템플릿에 반영하여 5개의 웹 스크립트들(Search, Select, Edit, Information, Action)을 생성한다. Action 스크립트를 제외한 나머지 스크립트들은 사용자 인터페이스로 각각 해당되는 웹 폼을 생성한다. 따라서 웹 응용 작성을 위한 시간과 노력을 크게 줄여 웹 응용의 생산성을 향상시킨다. WebGen은 상용 웹 응용 생성기들과는 다르게 <표 1>에서 보인 바와 같이 다음의 기능들을 추가로 제공한다.

1. 생성된 폼에 표현 가능한 정보도 관심의 대상인 엔티티 뿐만 아니라 이 엔티티와 직·간접적으로 연관된 모든 엔티티들을 포함한다.
2. 파라미터 교환 외에는 상호 독립적인 템플릿을 기반으로 하기 때문에 WebGen의 버전 관리가 용이하다.

향후, WebGen의 구성파일 작성 과정도 자동화 할 계획이다. 이는 응용 데이터베이스의 시스템 카탈로그로부터 테이블 간의 관계 정보를 추출하여 필요한 구성파일을 자동 생성함으로써 가능하다. 따라서 응용 프로그램 개발자는 구성파일을 작성하는 대신 원하는 테이블(및 엔티티)만 지정함으로써 응용에 필요한 스크립트들을 얻을 수 있게 된다. 또한, 모바일 시장의 급속한 확대로 모바일 응용의 생산성 향상이 요구되고 있으므로 WebGen을 모바일 응용을 자동 생성할 수 있는 MobileGen으로 확장할 계획이다.

참 고 문 헌

[1] DATAMONITOR, <http://www.datamonitor.com>.
 [2] Eum, D. and Minoura, T., Web-based database application generator. *IEICE Transactions on Information and Systems*, Vol.E86-D, No.6, June, 2003.
 [3] 음두현 고민정 강이지, 컴포넌트 기반 웹 데이터베이스 응용의 자동 생성, 한국정보처리논문지, 제11-D권, 제2호, 2004. 4.
 [4] AppGini3.1, <http://www.bigprof.com>.
 [5] FW Plus, <http://www.bmspro.net>.
 [6] ASP.NET Web : The Official Microsoft ASP.NET Site, <http://www.asp.net>.
 [7] PHPMaker4.02, <http://www.hkvstore.com>.
 [8] PostgreSQL, <http://www.postgresql.org>.
 [9] George Schlossnagle; *Advanced PHP Programming Sams*: 2003.
 [10] Danny Goodman; *JavaScript Bible John Wiley & Sons*: 2004.
 [11] Zhang, Towards Increasing Web Application Productivity, Proceedings of the ACM Symposium on Applied Computing, 2004.
 [12] Danny Goodman, *JavaScript Bible*, John Wiley & Sons, 2004.
 [13] George Schlossnagle, *Advanced PHP Programming*, Sams, 2003.
 [14] B. Doyle and C. Lopes, *Survey of Technologies for Web Application Development*, ACM Press, 2006.
 [15] Griss, M. and Pour, G., *Accelerating Development with Agent Components*, *IEEE Computer*, Vol.34, No.5, pp.37-43, 2001.

음 두 현



e-mail : dheum@duksung.ac.kr

1984년 서강대학교 전자공학과 (학사)

1987년 오레곤주립대학교 대학원

컴퓨터공학과 (석사)

1990년 오레곤주립대학교 대학원

컴퓨터공학과 (박사)

1991년~1992년 한국전자통신연구원 인공지능연구실
 선임연구원

1999년~2000년 오레곤주립대학교 컴퓨터과학과 교환교수

1992년~현 제 덕성여자대학교 컴퓨터공학부 교수

관심분야: 객체지향 시스템, 분산 및 모바일 응용, etc.