

모바일 데이터베이스 응용을 위한 스크립트 자동 생성기의 설계 및 구현[☆]

Design and Implementation of Automatic Script Generator for Mobile Database Applications

음 두 현*
Doohun Eum

요 약

무선 인터넷 기술의 발전과 함께 이용자 수가 증가하면서 모바일 데이터베이스 응용 프로그램의 수요가 급증하고 있다. 그러나 모바일 데이터베이스 응용에 필요한 사용자 인터페이스와 질의 처리 코드는 대부분 수작업으로 작성되기 때문에 모바일 응용의 생산성이 낮고 버전 관리에도 많은 시간과 노력이 필요하다. 본 논문에서는 모바일 데이터베이스 응용을 위한 스크립트 생성기인 MobileGen의 설계 및 구현을 기술한다. 생성된 스크립트들은 데이터베이스 연동 및 질의 처리 코드를 포함하고 간단한 폼 형태의 카드 또는 연계된 카드들의 집합인 데크를 사용자 인터페이스로 제공함으로써 모바일 데이터베이스 응용의 생산성을 향상시킨다. 생성된 스크립트들은 스크립트의 원형인 템플릿을 기반으로 하기 때문에 MobileGen의 버전 관리와 생성된 응용의 유지·보수를 용이하게 한다. 또한, MobileGen은 상용 스크립트 생성기들에 비해 관심의 대상인 엔티티 외에 이 엔티티와 직·간접적으로 연관된 모든 엔티티들에 대한 정보를 한 데크에 제공한다.

ABSTRACT

The demand for mobile database applications has been rapidly increased with the growth of mobile users and the development of wireless Internet technology. But the productivity of mobile applications is low and it takes much time to manage the versions of applications because the user interface and query processing code of applications is manually written. In this paper, we describe the design and implementation of the MobileGen that is a script generator for mobile database applications. The generated scripts enhance mobile application productivity by providing the code for operating with a database and processing user queries. Each script provides a corresponding deck that is a set of related cards as user interface. The MobileGen supports easy version management of generated applications and the MobileGen itself because it is based on the templates that are frameworks for scripts. Moreover, the MobileGen provides not only the interested entity but also the entities that are related directly and indirectly with the interested entity unlike the most commercial mobile script generators.

□ KeyWords : Mobile Script Generators, Mobile Applications, Mobile Application Productivity, Mobile Database Applications, Wireless Internet, 모바일 스크립트 생성기, 모바일 응용, 모바일 응용의 생산성, 모바일 데이터베이스 응용, 무선 인터넷

1. 서 론

최근 발표된 IDC사의 조사에 따르면 이동 통신 가입자 수는 2005년 20억 명에서 2010년에는 31억 명 이상으로 증가할 것이라고 한다. 또한, 모바일 응용 시장은 2005년 12억 달러에 달했으나, 23%의 연간 성장률을 보여 2010년에는 총 35억 달러에 이를 것으로 예측하고 있다. 이렇듯 세계

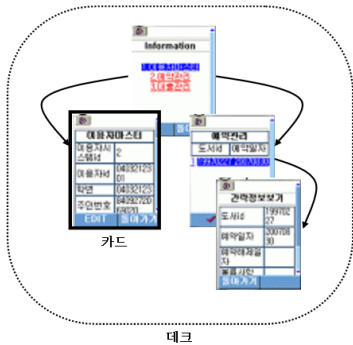
* 정 회 원 : 덕성여자대학교 컴퓨터공학부 교수
dheum@duksung.ac.kr

[2008/08/13 투고 - 2008/8/21 심사(2008/12/29 2차 - 2009/01/15 3차) - 2009/01/20 심사완료]

☆ 본 논문은 2008년 덕성여자대학교 교내 연구비 지원으로 연구되었음

적으로 모바일 시장의 규모는 이용자 수에 비례하여 커지고 있으며 그만큼 모바일 응용에 대한 생산성 향상도 요구되고 있다. 그러나 현재 모바일 응용의 작성은 대부분 수작업에 의존하고 있으며 작성 과정의 자동화 기술은 초보적인 단계에 있다[1, 2, 3].

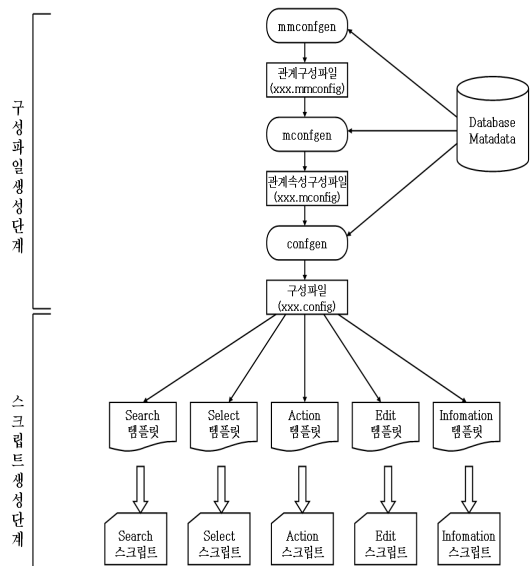
본 논문에서 소개하는 MobileGen은 모바일 스크립트 자동 생성기로서 모바일 응용에 필요한 그림 1의 카드(card)와 데크(deck)로 구성되는 인터페이스와 이 인터페이스를 통해 이루어지는 질의를 데이터베이스와 연동하여 처리하는 모바일 스크립트들을 생성한다. 카드는 텍스트 필드나 버튼 등으로 구성되는 간단한 폼 정보를 담고 데크는 상호 연결된 카드들의 집합으로서 모바일 기기의 전송 단위를 말한다. 사용자가 카드의 버튼을 클릭하면 데크 내의 해당 카드로 이동된다.



(그림 1) MobileGen이 제공하는 사용자 인터페이스

그림 2는 MobileGen을 이용한 모바일 스크립트의 생성 과정을 나타낸 것이다. 생성 과정은 응용 데이터베이스 내의 각 테이블에 대해 크게 구성파일 생성 단계와 스크립트 생성 단계로 나누어지며 mmconfgen, mconfgen, confgen, scriptgen 도구들을 사용하여 진행된다. 구성파일 생성 단계는 응용 데이터베이스의 테이블간의 관계와 속성들을 분석하여 구성파일을 작성한다. 속성은 카드나 데크를 구성하는 필드들의 필드명, 타입, 길이 등을 말한다.

mmconfgen은 응용 데이터베이스의 카탈로그 정보(메타 데이터)를 기반으로 추출한 테이블간의 관계를 관계구성파일 템플릿에 반영하여 각 테이블과 일대일, 다대일 관계 또는 다대일, 다대다 관계인 테이블들에 대한 관계 정보를 담은 관계구성파일(xxx.mmconfgen)을 생성한다. 템플릿은 생성될 파일의 기본 틀을 제공한다. mconfgen은 생성된 관계구성파일을 관계속성구성파일 템플릿에 반영하여 테이블들의 관계에 대한 정보와 간략한 속성들을 담은 관계속성구성파일(xxx.mconfgen)을 생성한다. 마지막으로 confgen은 관계속성구성파일을 구성파일 템플릿에 반영하여 세부적인 속성들을 담은 구성파일(xxx.confgen)을 생성한다.



(그림 2) MobileGen의 스크립트 생성 과정

스크립트 생성 단계는 scriptgen 도구를 통해 내장된 스크립트 템플릿(Search, Select, Edit, Information, Action)을 기반으로 구성파일의 내용을 반영하여 테이블 당 5개의 PHP 모바일 스크립트들(Search, Select, Edit, Information, Action)을 생성한다. 스크립트들은 데이터베이스 연동 및 질의

처리 코드를 포함하며 Action 스크립트를 제외한 나머지 스크립트들은 각각 카드 또는 데크를 사용자 인터페이스로 클라이언트에게 제공한다.

이러한 MobileGen의 스크립트 생성 과정은 전 자동으로 진행되어 모바일 응용의 생산성 향상을 지원한다. 예를 들어, 5개의 테이블들로 구성되는 응용 데이터베이스에 대해 MobileGen은 각 테이블 당 5개의 스크립트들을 테이블명만 주면 자동 생성하여 총 25개의 스크립트들을 생성한다. 또한, MobileGen은 템플릿을 기반으로 하기 때문에 데이터베이스의 구조 변경이나 스크립트들의 개별적인 변경에 대하여 템플릿 파일의 내용만 수정하면 스크립트들을 신속하게 재생산 할 수 있어 MobileGen 자체와 생산된 응용의 용이한 버전 관리를 지원한다.

MobileGen과 비교할만한 상용 모바일 스크립트 생성기들인 MobileDev[4], WINDEV Mobile[5], MobiForms[6], VisualStudio.NET[7] 등은 카드 또는 데크의 생성이나 데이터베이스 연동 부분이 대부분 반자동으로 이루어지기 때문에 생산성 지원이 미비하고 템플릿 기반이 아니기 때문에 코드 수정에 따른 파급 효과가 커 버전 관리가 쉽지 않다. 여기서 반자동이란 인터페이스(카드, 데크)의 구성과 데이터베이스의 연결 작업이 수작업인 것을 말한다. 수작업인 이유는 개발자에 의한 인터페이스 디자인과 정확한 테이블 연결을 위한 것이나 생성기가 표준 디자인을 제공하고 연결 테이블을 자동 감지할 수 있다면 그만큼 모바일 응용의 생성이 신속하고 용이할 것이다. 본 논문에서는 사용자가 현재 관심을 두고 있는 기준에 되는 엔티티를 닷 엔티티(anchor entity)라 하는데, MobileGen은 상용 모바일 스크립트 생성기들과 달리 닷 엔티티 외에 이 엔티티와 일대일, 다대일 관계의 엔티티들(첨부 엔티티, appended entity)을 포함하며 닷 엔티티와 일대다, 다대다 관계의 엔티티들(확장 엔티티, expanded entity)과 함께 이 확장 엔티티들과 일대일, 다대일 관계의 엔티티들(확장-첨부 엔티티, expanded-appended entity)을 한

데크에 제공한다. 상용 모바일 스크립트 생성기들이 제공하는 카드 또는 데크는 대부분 닷 엔티티만을 제공한다.

2절에서는 MobileGen의 개요를 설명하고 3절에서는 MobileGen과 상용 모바일 스크립트 생성기들의 기능을 비교한다. 4절에서는 MobileGen의 구현을 간략히 설명하고 마지막 절에서는 본 논문의 결론을 정리한다.

2. MobileGen의 개요

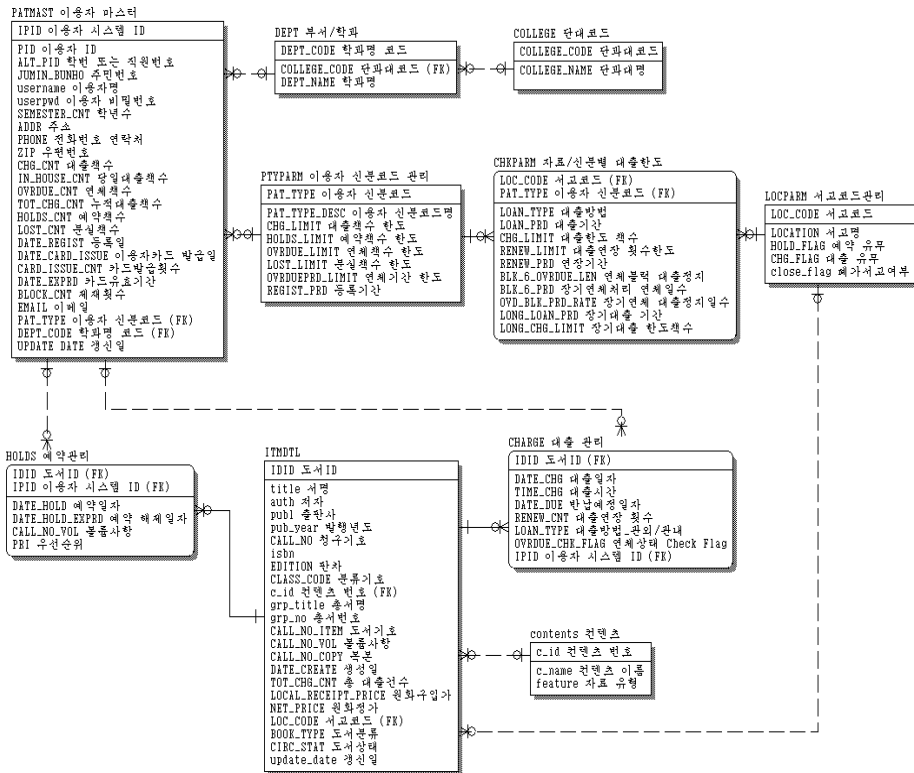
본 절에서는 MobileGen의 개요를 설명한다. 모바일 응용 개발자는 MobileGen의 구성파일 생성 단계와 스크립트 생성 단계를 거쳐 필요한 모바일 스크립트들을 생성한다. 구성파일 생성 단계는 응용 데이터베이스의 각 테이블에 대해 테이블간의 관계와 속성들 간의 관계를 정의하는 구성파일을 생성한다. 스크립트 생성 단계는 내장된 템플릿에 생성된 구성파일의 내용을 반영하여 테이블 당 5개의 모바일 스크립트들(Search, Select, Edit, Information, Action)을 생성한다. 5개의 모바일 스크립트들은 사용자 인터페이스로 4개의 카드 또는 데크를 제공하고 이들을 통한 사용자의 검색, 삭제, 삽입, 갱신 질의를 수행하는 코드를 제공한다.

그림 3은 덕성여자대학교 도서관에서 사용하고 있는 DISCOVER 응용 데이터베이스의 ER 다이어그램 중 대출, 검색, 컨텐츠 부분을 보인 것이다. MobileGen에서 닷 엔티티란 ER 다이어그램 중 기준이 되는 엔티티로서 생성될 스크립트 및 데크(카드)는 닷 엔티티를 중심으로 생성된다. 이용자 마스터 엔티티를 닷 엔티티로 설정할 경우, 이용자 마스터 엔티티는 예약관리, 대출관리 엔티티와 일대다의 관계를 가지고 부서/학과 엔티티, 이용자 신분코드 관리 엔티티와는 다대일의 관계를 가진다.

그림 4는 DISCOVER 응용 데이터베이스의 이용자 마스터 엔티티를 닷 엔티티로 했을 때 그림

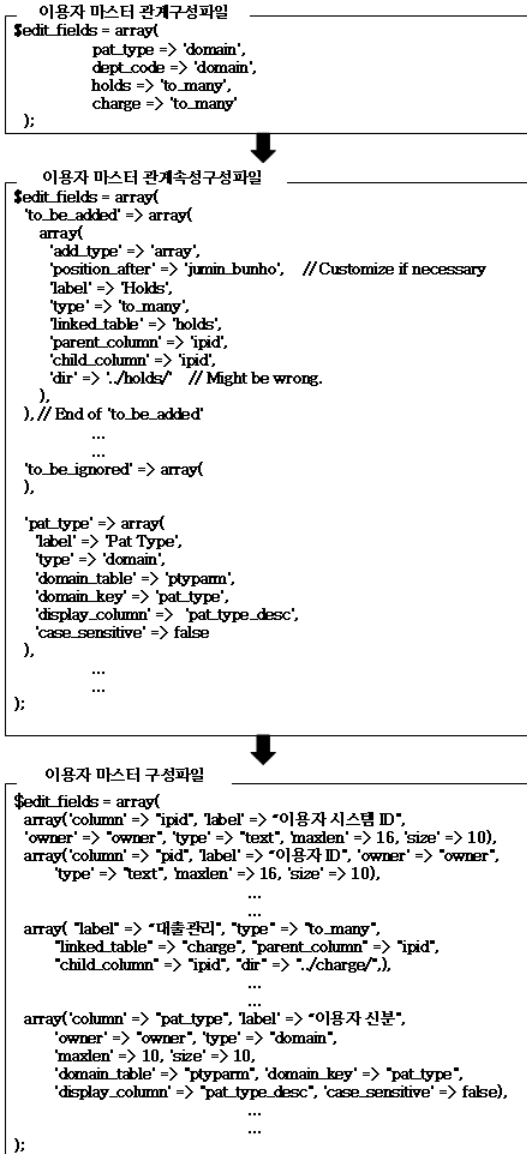
2의 구성파일 생성 단계에서 mmconfgen, mconfgen, confgen 도구들에 의해 자동 생성되는 파일들의 내용 중 일부를 보인 것이다. 사용자 마스터 관계구성파일 중 Edit 스크립트의 생성을 위한 \$edit_fields는 PHP 배열이다. 이 \$edit_fields에서 그림 3의 PTYPARM과 DEPT 엔티티를 지정하는 pat_type과 dept_code는 PATMAST 닷 엔티티와 일대일 관계를 나타내는 domain으로 설정되며 HOLDS와 CHARGE 엔티티를 지정하는 holds와 charge는 일대다 관계를 나타내는 to_many로 설정된다. 사용자 마스터 관계속성구성파일 중 \$edit_fields는 관계구성파일의 \$edit_fields를 구체화 시킨다. domain 타입을 갖는 pat_type은 Pat Type이라는 레이블명을 가지고 도메인 테이블명은 ptyparm이며 도메인 키는 pat_type, 표시된 컬럼(display_column)은 pat_type_desc로 정의된다. to_many 타입을 갖는 holds는 Holds라는 레이블명

이 붙여지고 부모 컬럼(parent_column)과 자식 컬럼(child_column)은 PATMAST 닷 엔티티의 키 필드인 ipid로 정의되며 현재 파일의 위치(dir)는 ../holds/의 경로로 설정된다. 사용자 마스터 구성파일 중 \$edit_fields는 관계속성구성파일의 \$edit_fields에 의해 관계가 설정된 엔티티들(pat_type, dept_code, holds, charge)에 대해 파라미터의 최대크기(maxlen)와 텍스트 필드 등의 크기를 추가한다. pat_type 배열의 속성들을 보면 관계속성구성파일의 내용과 함께 파라미터의 최대 크기는 10, 텍스트 필드의 크기는 10으로 정의된다. 또한, 닷 엔티티인 사용자 마스터 엔티티의 컬럼에 대한 속성들도 정의가 되는데 \$edit_fields의 첫 번째 열의 컬럼명은 ipid이며 모바일 브라우저에 표시될 레이블명도 ipid이다. Search, Select, Information 스크립트들의 생성을 위한 \$search_fields, \$select_fields, \$info_fields 배열들도



(그림 3) DISCOVER 응용 데이터베이스에 대한 ER 다이어그램(일부)

비슷한 방법으로 정의되어 구성파일을 생성한다.



(그림 4) DISCOVER 데이터베이스에서 구성파일 생성 단계로부터 생성된 파일들

그림 5는 DISCOVER 데이터베이스에 대해 그림 2의 스크립트 생성 단계에서 scriptgen 도구에 의해 자동 생성된 스크립트들의 동작 과정을 보인 것이다. 생성된 Search 스크립트에 의해 한 개

의 카드로 구성되는 Search 데크(①)가 첫 화면으로 브라우저에 표시된다. Search 데크는 닷 엔티티인 사용자 마스터 엔티티에 관한 정보를 검색할 수 있게 구성된다. 사용자는 원하는 항목에 검색 조건(파라미터)을 입력하고 버튼을 누른다. 버튼을 누르면 입력된 파라미터를 Select 스크립트에 전달하는 이벤트가 발생되고 검색 조건을 만족하는 레코드들이 데이터베이스에서 검색되어 Select 스크립트가 생성하는 Select 데크(②)를 통해 사용자에게 제공된다. 검색 조건이 없을 경우는 모든 사용자 마스터 레코드들이 제공된다.

카드들은 무선 단말기의 작은 LCD 창에 디스플레이 될 수 있는 단위로 설계되었다. 즉, 첫 화면에 표현되는 카드에는 데이터베이스 테이블에 정의된 컬럼들 중 제일 앞의 두 컬럼명만 보이고 서브메뉴를 두어 간략정보보기(②-1, 테이블의 나머지 컬럼 정보를 제공, 삭제(②-2, 선택된 레코드 삭제), 삽입(②-3), 돌아가기를 할 수 있도록 구성된다. 또한, 한 페이지에 표시되는 레코드의 개수는 8로 설정된다.

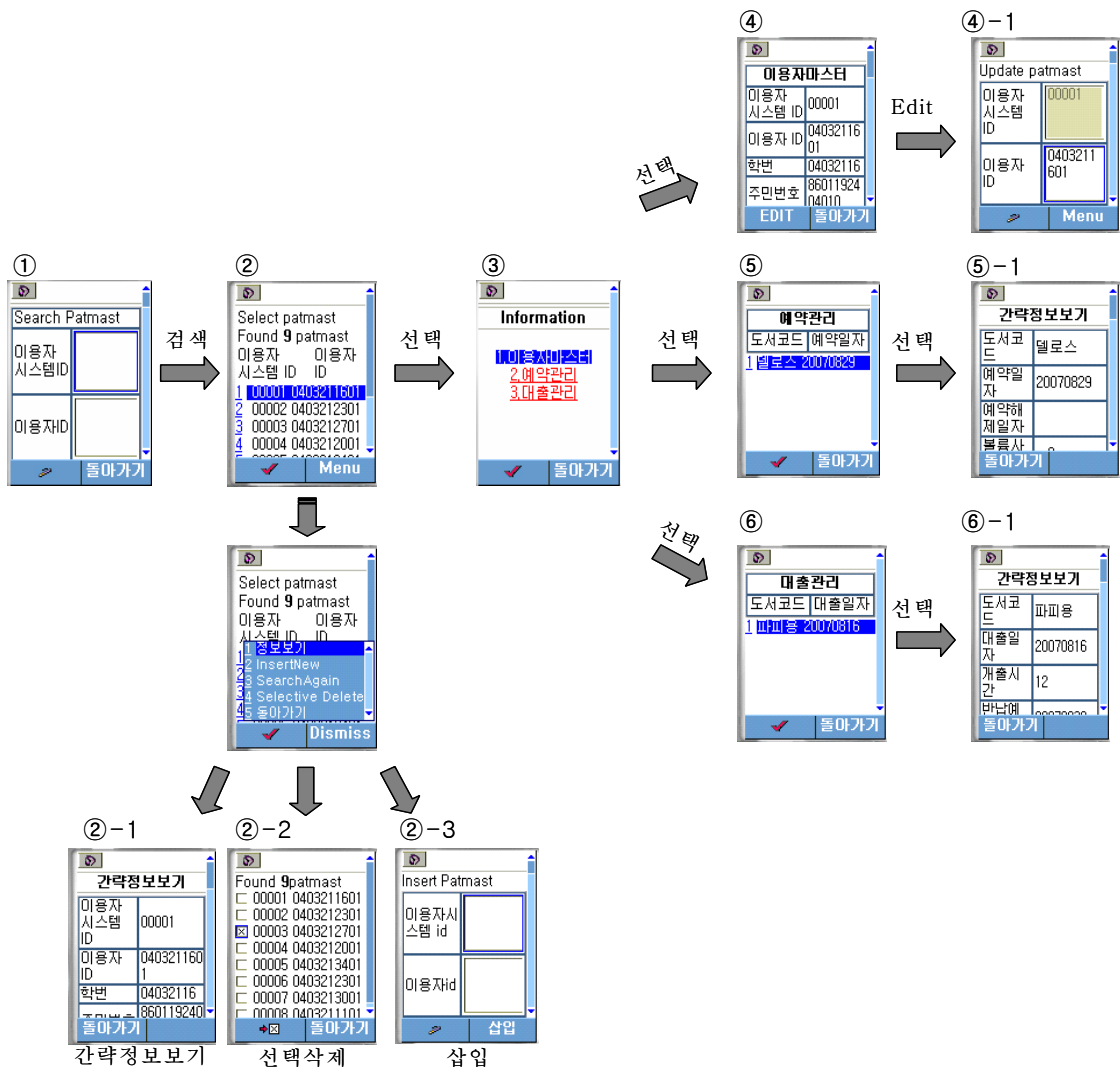
Information 스크립트가 제공하는 Information 데크(③)는 Select 데크에서 선택된 레코드에 관한 정보를 표시한다. Information 데크의 첫 번째 화면은 닷 엔티티(사용자 마스터)와 확장 엔티티(예약관리, 대출관리)에 대해 링크를 포함하는 카드다. 첫 번째로 링크된 카드(④)는 닷 엔티티인 사용자 마스터 엔티티와 첨부 엔티티인 부서/학과와 사용자 신분코드 관리 엔티티에 관한 정보 카드이며 링크된 나머지 카드들(⑤, ⑥)은 확장 엔티티인 예약관리와 대출관리 엔티티 및 확장-첨부 엔티티인 도서 엔티티에 관한 정보 카드다. 사용자가 ③번 카드에서 닷 엔티티인 사용자 마스터 엔티티를 선택하면 Select 데크에서 선택되었던 사용자 마스터 엔티티에 관한 자세한 정보를 볼 수 있고(④), 여기서 Edit 버튼을 누르면 Edit 스크립트가 생성하는 Edit 데크(④-1)로 연결된다. 예약관리, 대출관리를 선택하면 Select 데크(②)와 마찬가지로 두 개의 컬럼 명만을 제시하는 카드(⑤,

⑥)와 함께 자세한 정보 보기 카드(⑤-1, ⑥-1)가 제공된다.

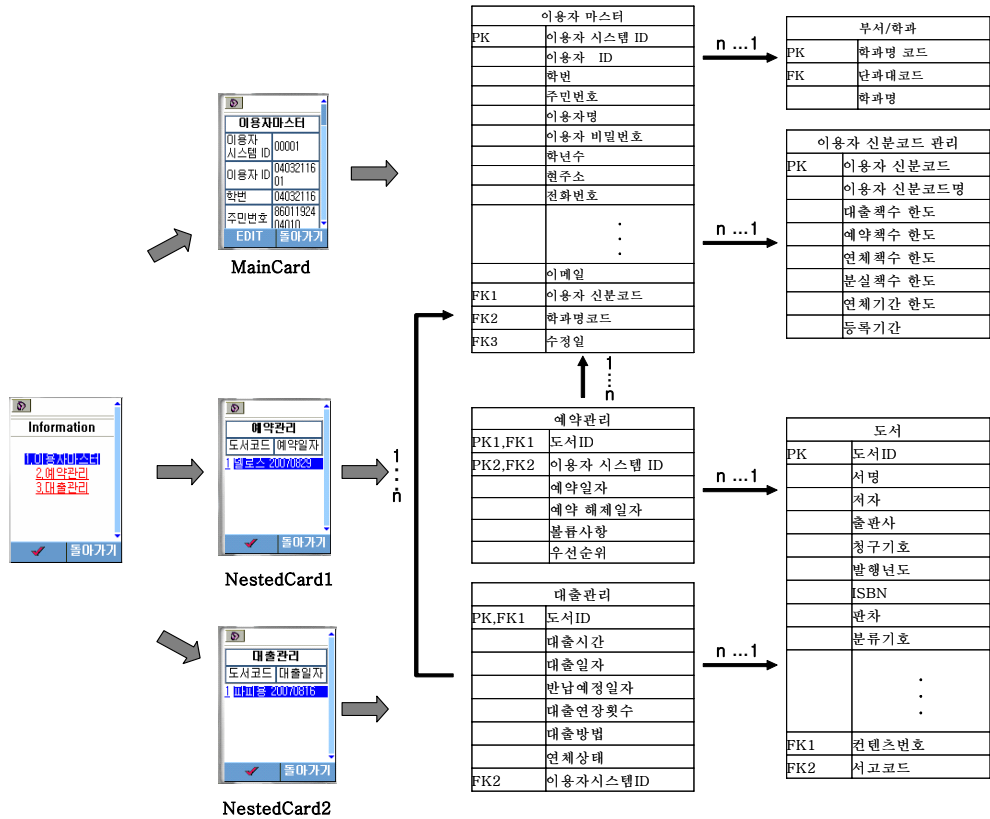
Edit 데크(②-3, ④-1)는 레코드의 수정이나 삽입 시 사용되며 사용자가 Select 데크의 메뉴에서 Insert New를 선택하거나(②-3) Information 데크에서 Edit 데크로 이동시 갱신을 위해 사용된다(④-1). 닷 엔티티의 모든 컬럼은 수정 가능한 필드로 표시되며 확장 엔티티에 대한 부분은 링크(SHOW) 버튼을 통해 표시된다. 닷 엔티티와 첨부

엔티티도 링크(VIEW 버튼, SELECT 버튼)를 걸어 나타내고 사용자가 값을 선택할 수 있도록 한다.

그림 6은 그림 5의 Information 데크(③~⑥)가 커버하는 엔티티들을 표현한 것이다. 첫 번째 카드인 MainCard는 닷 엔티티인 사용자 마스터 엔티티 자신과 함께 닷 엔티티의 첨부 엔티티인 부서/학과 엔티티와 사용자 신분코드 관리 엔티티에 대한 정보를 제공한다. 부서/학과 엔티티와 사용자 신분코드 관리 엔티티는 닷 엔티티와 다대일



(그림 5) MobileGen으로 생성한 스크립트들의 동작 과정



(그림 6) Information 데크의 엔티티 커버리지

관계이다. 두 번째 카드인 NestedCard1은 닷 엔티티의 확장 엔티티 정보가 제공된다. 사용자가 예약한 도서에 대한 정보가 리스트 되며 간략 정보를 통해 자세한 정보를 제공하고 가장 마지막 컬럼에는 예약관리 엔티티와 다대일 관계에 있는 도서 엔티티(확장-첨부 엔티티)의 정보까지 제공된다. 세 번째 카드인 NestedCard2는 닷 엔티티와 일대다의 관계인 대출관리 엔티티(확장 엔티티)에 대한 정보와 대출관리 엔티티와 다대일의 관계인 도서 엔티티(확장-첨부 엔티티)에 대한 정보를 제공한다. 이와 같이, Information 데크는 닷 엔티티와 직접적으로 연관된 첨부 엔티티와 확장 엔티티의 정보뿐만 아니라 간접적으로 연관된 확장-첨부 엔티티의 정보까지 제공함으로써 응용 데이터베이스의 테이블들에 대한 높은 커버

리지를 지원한다.

3. MobileGen의 기능 비교

본 절에서는 MobileGen과 상용 모바일 스크립트 생성기들의 기능을 비교한다. 표 1은 MobileGen과 상용 모바일 스크립트 생성 환경인 MobileDev[4], WINDEV Mobile[5], MobiForms[6], VisualStudio.NET[7]의 기능을 비교한 것이다. 선택된 생성기들은 검색 엔진에 의해 검색된 모바일 응용의 생성을 지원하는 것들 중, 논문과 기술 문서 등에서 인용이 많이 된 생성기들이다. 또한, 비교 항목들은 자동화 정도를 나타내는 항목(스크립트 생성, 데이터베이스 연동, 한 데크에 표현 가능한 엔티티)[8], 유지·보수에 관한 항목(생성

기의 버전 관리, 생성된 스크립트의 유지·보수), 정성적인 종합 평가 항목(생산성) 등을 포함한다.

MobileDev는 SpeedWare사에서 개발한 WAP 기반의 인터넷 응용 개발 환경이다. MobileDev는 ASP, Perl, JSP 기술을 지원하며 Windows 환경에서, 질의 처리 도구인 Query Wizard를 사용하여 WML, HDML 코드로 작성된 카드 또는 데크를 생성한다. 또한, 개발자는 GUI 응용 mapper 도구를 사용하여 WAP 응용 컴포넌트들 간의 관계를 파악할 수 있다. 그러나 데이터베이스와 연동하는 질의 처리 스크립트의 생성이 반자동이기 때문에 웹 응용의 높은 생산성을 지원하지는 못한다. 여기서 반자동이란 데이터베이스와 연결하는 작업이 수작업인 것을 의미한다. MobileDev에서, 데이터베이스 엔티티들 간에 일대일, 일대다, 다대일, 다대다 관계의 설정이 가능하지만 하나의 데크에는 닷 엔티티만이 제한적으로 제공된다. MobileDev는 수정이 필요할 경우, GUI 응용 mapper 도구를 이용해 컴포넌트들 간의 관계 및 기본적인 내용의 변경은 쉽지만 코드 레벨에서는 개별적으로 스크립트들을 수정해야 하기 때문에 프로그램의 버전 관리나 생성된 스크립트들의 유지·보수가 쉽지 않다.

WINDEV Mobile은 포켓 PC, 스마트 폰 등을 위

해 PcSoft사에서 개발한 응용 개발 환경이다. GUI 도구인 monitor를 사용하여 모바일 응용을 위한 프로젝트를 정의하고 데이터 파일을 묘사하는 analysis 도구와 연결시켜 XML 코드를 반자동으로 생성한다. 직접적인 질의 처리 없이 다이어그램을 이용하여 XML 코드의 생성이 가능하나 프로젝트에 생성 및 설정을 수작업으로 해줘야 하기 때문에 생산성 지원은 미흡하다. WINDEV Mobile은 버전 관리 시 새로운 특징만을 추가 지원한다는 점에서 프로그램 자체의 버전 관리는 용이한 편이다. WINDEV Mobile은 생성하는 응용 부분을 멀티 프로젝트 관리와 프로젝트에 대한 폴더 관리로 나누어 하기 때문에 생성된 응용에 대한 유지·보수도 잘 지원하는 편이다. 그러나 WINDEV Mobile에서 데이터베이스에 대한 접근은 가능하지만 하나의 데크에 표현 가능한 엔티티는 닷 엔티티로 제한적이다.

MobiForms는 자바 기반의 모바일 데이터베이스 응용 개발 환경이다. MobiForms는 모바일 데이터베이스 응용과 MobiForms의 실행을 지원하는 XML의 특별 버전인 FML(Forms Mark-up Language) 코드를 반자동으로 생성한다. MobiForms는 Oracle, Sybase 등의 데이터베이스 서버를 지원하지만 데이터베이스의 동기화 문제

(표 1) 상용 모바일 스크립트 생성기와의 기능 비교

		MobileGen	MobileDev	WINDEV Mobile	MobiForms	VisualStudio .NET
자동화 항목	스크립트 생성	지원	지원	지원	지원	지원
	데이터베이스 연동	자동	반자동	반자동	반자동	반자동
	한 데크에 표현 가능한 엔티티	닷, 첨부, 확장, 확장-첨부 엔티티	닷 엔티티	닷 엔티티	닷 첨부 엔티티	닷 엔티티
생성기의 버전 관리		+++	++	+++	+++	++
생성된 스크립트에 대한 유지·보수		+++	++	+++	+++	++
생성 스크립트언어		PHP, WML	WML, HDML	XML	FML	WML, cHTML, HTML
운영체제		Linux, Windows	Windows	Windows	Windows	Windows
생산성		+++	++	++	++	++

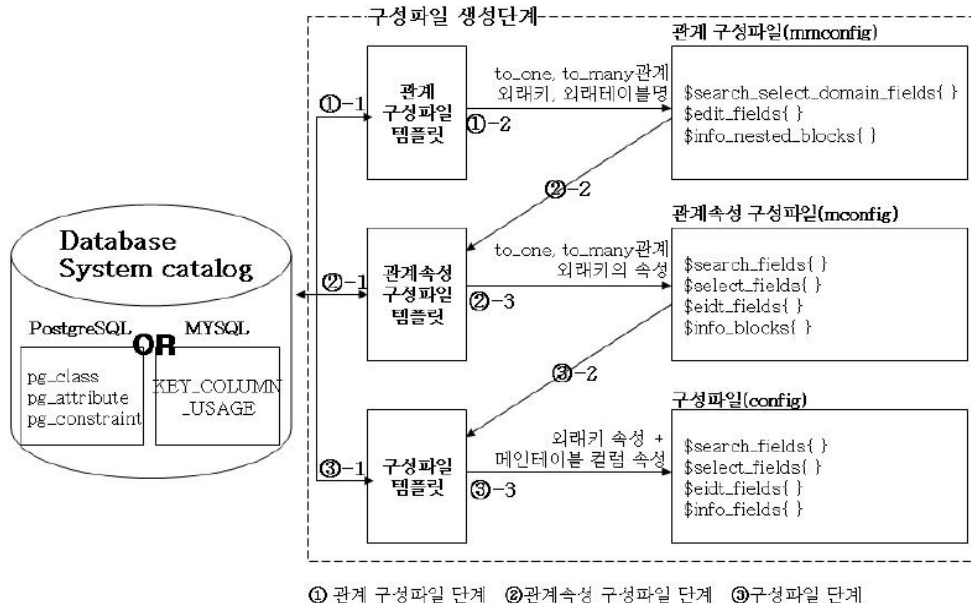
를 지원하지는 않는다. 데이터베이스와의 연동은 사용자 인터페이스로 제공된 폼을 이용하여 설정할 수 있지만 반자동이기 때문에 모바일 응용의 생산성 지원은 미약한 편이다. 또한, 하나의 데크에는 닷 엔티티와 첨부 엔티티만을 제공한다. 그러나 개발자는 GUI 환경에서 갱신, 삭제, 삽입 질의 처리 도구를 통해 효과적인 코드 생성과 변경을 할 수 있어 생성된 스크립트의 유지·보수는 용이하다. Mobiforms는 무선 기기의 데이터 보존을 위한 갱신과 보호를 지원하고 이전 버전에 대한 통합 관리로 인해 프로그램 자체의 버전 관리도 용이하다.

VisualStudio.NET은 스크립트 생성에 필요한 GUI 요소들을 선택적으로 구성하여 WML, HTML, cHTML의 코드를 생성해 줄 수 있는 개발 환경이다. VisualStudio.NET은 CLR(Common Language Runtime)을 포함한 닷넷 프레임워크에서 비주얼 베이직, 비주얼 C++, 비주얼 C# 등의 다양한 프로그래밍 언어를 지원한다. 모바일 스크립트는 개발자의 선택에 따라 생성되며 하나의 데크에는 MobileDev나 WINDEV Mobile과 같이

닷 엔티티만 표현 가능하다. VisualStudio.NET도 데이터베이스를 통한 질의 처리 스크립트의 생성이 반자동이기 때문에 생산성 지원은 MobileGen에 비해 미흡한 편이다. VisualStudio.NET은 프로젝트 파일 단위로 작업을 관리하기 때문에 프로젝트 파일에 대한 수정을 통해 생성된 응용의 유지·보수는 쉬운 편이다.

4. MobileGen의 구현

본 절에서는 MobileGen의 구성파일 생성 단계와 스크립트 생성 단계를 자세히 설명한다. 4.1절에서는 구성파일의 생성과 스키마 정보의 추출 과정을 설명하고 4.2절에서는 스크립트 생성 과정과 카드 및 데크의 생성 알고리즘을 설명한다. MobileGen은 PHP5.2.3과 WML로 구현되었으며 PostgreSQL7.4.17[9] 또는 MySQL5.0.41[10]과 Apache2.0.52 서버를 사용한다.



(그림 7) MobileGen의 구성파일 생성 과정

4.1 구성파일의 생성

4.1.1 구성파일의 생성 과정

그림 7은 MobileGen이 데이터베이스 시스템 카탈로그로부터 메타 데이터 정보를 추출해 구성파일을 생성하는 세부 단계를 보인 것이다. 구성파일 생성 과정은 시스템 카탈로그 접근, 상위 구성파일 반영, 구성파일 생성의 세부 단계들을 거친다. 그림 7에서 관계구성파일 단계(①)는 상위 구성파일이 존재하지 않으므로, 시스템 카탈로그 접근과 구성파일 생성의 두 단계로 이루어진다.

관계구성파일 단계(①)는 MobileGen의 관계구성파일 템플릿을 기반으로 관계구성파일을 생성한다. ①-1은 관계구성파일 템플릿이 시스템 카탈로그에 접근하는 단계이다. 이 단계에서는 메인 테이블의 외래키, 외래키 테이블명, 외래키 테이블의 기본키 정보 등을 얻는다. 테이블 A에서 테이블 B로의 관계가 일대일, 다대일 관계이면 to_one 타입으로 지정되고 다대다, 일대다 관계이면 to_many 타입으로 지정된다. ①-2는 관계구성파일 생성 단계로, ①-1에서 얻은 정보를 이용하여 관계구성파일 템플릿을 기반으로 관계구성파일의 \$search_select_domain_fields, \$edit_fields, \$info_nested_blocks 배열을 구성한다. \$search_select_domain_fields는 to_one 관계의 외래키를 가진다. \$edit_fields는 외래키 타입이 to_one이면 외래키와 to_one 또는 domain 타입을 나타내고, 외래키 타입이 to_many이면 외래 테이블명과 to_many 타입을 나타낸다. \$info_nested_blocks는 외래 테이블명을 저장한다.

관계속성구성파일 단계(②)는 관계구성파일과 관계속성구성파일 템플릿을 기반으로 관계속성구성파일을 생성한다. ②-1은 관계속성구성파일 템플릿이 시스템 카탈로그에 접근하는 단계이다. 이 단계에서는 메인 테이블의 기본키, 외래키, 외래키 테이블명, 외래키 테이블의 기본키 정보 등을 얻는다. ②-2는 상위 구성파일인 관계구성파일을 반영하는 단계이다. 관계구성파일의

\$search_select_domain_fields 배열 값인 외래키, \$edit_fields 배열 값인 외래키, to-one 또는 domain 타입, 외래키 테이블명, to_many 타입, \$info_nested_blocks 배열값인 외래키 테이블명 등을 얻는다. ②-3은 ②-2에서 얻은 상위 구성파일과 카탈로그 정보를 합쳐 관계속성구성파일을 생성하는 단계이다. 관계구성파일의 \$search_select_domain_fields 배열 값과 카탈로그에서 얻어온 정보를 합쳐 외래키의 속성을 만들고 관계속성구성파일의 \$search_fields와 \$select_fields에 저장한다. 관계구성파일의 \$edit_fields 배열에서 파라미터를 얻어와 카탈로그에서 얻어온 정보와 합쳐 외래키의 속성을 만들고 \$edit_fields 배열 속성 중 타입이 to_many 또는 to_one이면 관계를 이루는 테이블명.mmconfig 파일의 디렉토리 상대 경로를 관계속성구성파일 \$edit_fields 배열의 변수(\$dir)에 저장한다. \$dir은 자동 생성된 5개의 스크립트에서 외래키의 테이블 경로로 사용된다. 관계속성구성파일의 \$info_blocks 배열은 main 또는 nested 타입으로 정의된다. main 타입은 \$search_fields나 \$select_fields의 내용을 가지며 nested 타입은 관계구성파일의 \$info_nested_blocks로부터 테이블명을 얻어 내포된 테이블의 카탈로그 정보와 내포된 테이블의 컬럼 속성 정보를 가진다.

구성파일 단계(③)는 관계속성구성파일과 구성파일 템플릿을 기반으로 구성파일을 생성한다. ③-1은 구성파일 템플릿이 카탈로그에 접근하는 단계이다. 이 단계에서는 타입별 제한 길이 정보를 얻고 SQL 질의를 사용하여 메인 테이블의 컬럼명과 타입 등을 얻는다. ③-2는 상위 구성파일인 관계속성구성파일을 반영 단계이다. 관계속성구성파일의 \$search_fields와 \$select_fields 배열 값인 외래키 정보와 \$edit_fields 배열 값인 외래키, 외래 테이블 정보, \$info_blocks 배열 값인 내포된 테이블 정보 등을 얻는다. ③-3은 ③-2에서 얻은 상위 구성파일과 카탈로그 정보를 합쳐 구성파일을 생성하는 단계이다. 관계속성구성파일의

\$search_fields와 \$select_fields 배열의 값과, 카탈로그와 SQL 질의를 통해 얻은 메인 테이블 정보를 합쳐 구성파일의 \$search_fields와 \$select_fields를 만든다. 관계속성구성파일의 \$edit_fields 배열 값과 메인 테이블 정보를 합쳐 구성파일의 \$edit_fields 배열을 구성하며 관계속성구성파일의 \$info_blocks 배열 값과 메인 테이블 정보를 합쳐 구성파일의 \$info_fields 배열을 구성한다.

4.1.2 스키마 정보의 추출

MobileGen은 데이터베이스가 PostgreSQL일때 pg_class, pg_constraint, pg_attribute 시스템 테이블들을 사용하고 MySQL일때는 INFORMATION_SCHEMA 중 KEY_COLUMN_USAGE 시스템 테이블을 사용하여 필요한 메타 데이터 정보를 추출한다. 본 절

에서는 PostgreSQL을 사용하는 경우를 설명한다.

MobileGen은 관계구성파일 템플릿과 관계속성구성파일 템플릿, 구성파일 템플릿에서, 스키마 정보를 얻기 위해 그림 8의 SQL 질의를 사용한다. PostgreSQL의 pg_class 테이블은 테이블들의 관계에 대한 종합적인 정보를 담고 있다. 즉, 사용자가 생성한 테이블명과 소유자, 튜플과 속성의 개수에 대한 내용을 저장한다. pg_attribute 테이블은 컬럼 정보를 담고 있으며 pg_constraint 테이블은 기본키, 단일키, 외래키에 관한 정보를 저장한다. MobileGen은 동일 테이블의 조인으로 생기는 컬럼명의 모호성을 해결하기 위해 별칭을 사용한다. t1은 메인 테이블, t2는 외래키 테이블이고, a1은 메인 테이블의 외래키, a2는 외래키 테이블의 기본키이다. \$table_name은 정보를 얻고자 하는 테이블명을 저장한다.

```

① SELECT a1.attname as fkey_in_main_table, t2.relname as foreign_table,
    a2.attname as pkey_in_foreign_table
FROM   pg_constraint c, pg_class t1, pg_class t2,
    pg_catalog.pg_attribute a1, pg_catalog.pg_attribute a2
WHERE  c.contype = 'f' and c.conrelid = t1.oid
    AND c.confrelid = t2.oid
    AND a1.attrelid = t1.oid AND a1.attnum = conkey[1]
    AND a2.attrelid = t2.oid
    AND a2.attnum = confkey[1] AND t1.relname = '$table_name';

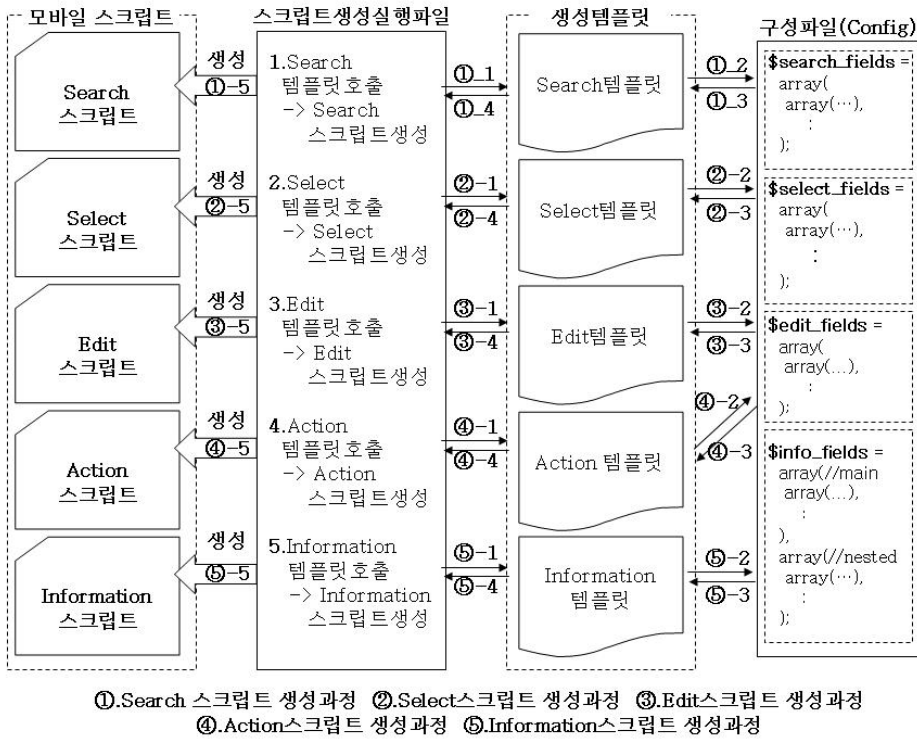
② SELECT a1.attname as my_column,
    t2.relname as other_table, a2.attname as fkey_in_other_table
FROM   pg_constraint c, pg_class t1, pg_class t2,
    pg_catalog.pg_attribute a2, pg_catalog.pg_attribute a1
WHERE  c.contype = 'f' and c.conrelid = t2.oid
    AND c.confrelid = t1.oid
    AND a2.attrelid = t2.oid AND a2.attnum = conkey[1]
    AND a1.attrelid = t1.oid
    AND a1.attnum = confkey[1] AND t1.relname = '$table_name';

③ SELECT a.attname as pkey
FROM   pg_constraint c, pg_class t1, pg_catalog.pg_attribute a
WHERE  c.contype = 'p' AND c.conrelid = t1.oid AND t1.oid = a.attrelid
    AND a.attnum = c.conkey[1] AND t1.relname = '$table_name';
    
```

(그림 8) PostgreSQL의 시스템 카탈로그 접근을 위한 SQL 질의문

그림 8에서 ①번 질의는 MobileGen이 테이블 A에서 테이블 B로의 관계가 to-one 타입일 때 사용된다. 한 개의 pg_constraint(c) 테이블에 pg_class(t1,t2) 테이블, pg_attribute(a1,a2) 테이블의 별칭을 주어 서로 다른 두 개의 테이블처럼 조인한다. pg_constraint(c) 테이블의 conrelid와 참조 관계에 있는 pg_class(t1) 테이블의 oid를 조인하고, pg_class(t1) 테이블의 oid와 참조 관계에 있는 pg_attribute(a1) 테이블의 attrelid를 조인하며, pg_attribute(a1) 테이블의 attnum과 참조 관계에 있는 pg_constraint(conkey[1])를 조인한다. pg_constraint(c), pg_class(t2), pg_attribute(a2)도 이와 같은 방법으로 조인한다. 조인된 where 절에 제약조건은 외래키(f)이고 \$table_name 테이블명을 갖는 열의 pg_attribute 테이블의 컬럼ID(메인 테이블의 외래키), pg_class 테이블의 테이블명, pg_attribute 테이블의 컬럼ID(외래키 테이블의 기본키) 등을 추출한다. 즉, ①은 테이블 A에서 테이블

B로의 타입이 to_one인 경우 메인 테이블의 외래키, 외래키 테이블명, 외래키 테이블의 기본키 정보를 추출한다. ②번 질의는 테이블 A에서 테이블 B로의 관계가 to_many 타입일 때 사용된다. ②번 질의문은 컬럼에 대한 별칭만 다를 뿐 동작 과정은 ①과 같으며 메인 테이블의 외래키, 외래키 테이블명, 외래키 테이블의 기본키 정보를 추출한다. ③번 질의는 메인 테이블의 기본키를 추출한다. 메인 테이블의 경우만 비교하므로 한 개의 pg_constraint(c) 테이블, pg_class(t1) 테이블, pg_attribute(a) 테이블 등이 사용된다. 동작 과정은 ①번 질의와 유사한 과정을 거쳐 pg_constraint(c)와 pg_class(t1), pg_attribute(a)를 조인한다. 조인된 where 절에 제약조건은 기본키(p)이고 \$table_name의 테이블명을 갖는 열의 pg_attribute의 컬럼ID(메인 테이블의 기본키)를 추출한다.



(그림 9) 스크립트 생성의 세부 단계

```

<card id=top_list>
    $info_block의 title를 표시
</card>

foreach ( $j=$info_fields의 서브 배열수 만큼 반복(인덱스) )
{
    switch(table type)
    {
        case 'main':
            <card id=card+인덱스(card $j)>
                for (main_block의 필드 수만큼 반복)
                {
                    main_block필드의 column, label, subtype을 얻음
                    subtype에 따라 출력문 생성.
                }
            </card>

        case 'nested':
            <card id=card+인덱스(card $j)>
                reference table의 정보를 얻어오는 쿼리문 생성
                if(필드배열의 출력 수 (<=2) 2개의 컬럼이름 출력
                for (nested_block의 필드 수만큼 반복){
                    nested_block필드의 column, label, subtype을 얻음
                    subtype에 따라 출력문 생성.
                }
                페이지 계산로직 출력
            </card>

            <card id=card+인덱스__ $title(card $j_ $title)>
                for(필드의 배열 수만큼 반복)
                {
                    각 배열의 'label'과 label 값을 갖는 변수 출력
                }
            </card>
        }
    }
}

```

(그림 10) 카드 및 데크의 생성 알고리즘

4.2 스크립트의 생성

4.2.1 스크립트의 생성 과정

그림 9는 스크립트 생성 과정에서 템플릿 호출, 구성파일 호출, 구성파일 반환, 템플릿 반환, 스크립트 생성으로 진행되는 세부 단계들을 보인 것이다. 그림 9에서 스크립트생성실행파일은 내장된 생성 템플릿들(Search, Select, Edit, Information, Action)

을 기반으로 5개의 모바일 스크립트들(Search, Select, Edit, Information, Action)을 생성한다.

①은 Search 스크립트의 생성 과정을 나타낸 것이다. ①-1은 Search 템플릿 호출 단계로서 테이블명, 디렉터리 상대 경로를 파라미터로 Search 템플릿에 보낸다. ①-2는 구성파일의 \$search_fields 값을 얻기 위해 호출하는 단계이고 ①-3은 구성파일의 \$search_fields 값을 Search 템플릿으로 반환

하는 단계이다. 구성파일의 \$search_fields는 닷 엔티티의 컬럼명, 타입, 사이즈, 파라미터의 최대 크기 등의 정보를 가진다. ①-4는 Search 템플릿 반환 단계로서 Search 템플릿은 구성파일에서 얻어진 \$search_fields 값을 이용하여 타입에 따른 유효성 확인, 컬럼명에 따른 레이블 지정, 사이즈에 따른 텍스트의 필드 크기 지정, 파라미터의 최대 크기에 따른 입력 길이 제한 등을 위한 코드를 생성하여, 이것을 스크립트생성실행파일로 반환한다. ①-5는 Search 스크립트 생성단계로 ①-4에서 반환된 Search 스크립트 코드들을 담은 Search 스크립트 파일을 생성한다. 나머지 4개의 스크립트들 (Select, Edit, Action, Information)도 이와 유사한 과정을 거쳐 생성된다.

4.2.2 카드 및 데크의 생성 알고리즘

MobileGen은 무선 단말기의 적은 메모리 용량에 맞는 데이터 처리를 위해 Information 스크립트를 생성하는 Information 템플릿에서 표시될 정보를 카드 단위로 나누고 이 카드들을 데크로 묶는 작업을 수행한다. 그림 10은 Information 템플릿의 카드 및 데크의 생성 알고리즘이다. information 템플릿은 1개의 top_list와 1개 이상의 card+인덱스, card+인덱스_\$title 형식으로 card id를 지정한다. card id는 각 카드를 구분하는 식별자이며 card id를 구성하는 인덱스는 \$info_fields 배열의 서브배열을 구분하기 위한 값이다. 0부터 서브배열의 수까지가 인덱스로 사용된다. top_list 카드는 \$info_fields의 서브배열 title을 표시하며 title 클릭시 card+title의 인덱스 카드를 브라우저에 표시한다. \$info_fields의 서브배열 타입이 main이면 서브배열의 모든 인자를 card+title의 인덱스 카드에 나타내고 nested이면 키가 되는 2개의 컬럼만 card+title의 인덱스에 보이고 전체 컬럼은 card+인덱스_\$title 카드에 표시한다. card+인덱스 카드의 생성은 구성파일의 \$info_fields 내부 배열 수에 따라 php의 foreach 문을 이용해 반복된다.

5. 결론

본 논문에서 소개하는 MobileGen은 모바일 데이터베이스 응용을 위한 스크립트 생성 도구이다. MobileGen을 덕성여자대학교 도서관에서 사용하고 있는 DISCOVER와 같은 실용 데이터베이스에 적용하면 구성파일 생성 단계, 스크립트 생성 단계를 거쳐 테이블 당 5개의 모바일 스크립트들 (Search, Select, Edit, Information, Action)을 생성한다. 생성된 스크립트들은 간단한 폼 형태의 카드와 상호 연결된 카드들의 집합인 데크를 사용자 인터페이스로 생성하고 데이터베이스와 연동하여 카드를 통해 이루어지는 질의를 처리하는 코드들을 생성한다. 표 1에서 보인 바와 같이 상용 모바일 스크립트 생성기들과 비교한 MobileGen의 장점을 요약하면 다음과 같다.

1. 모바일 데이터베이스 스크립트들을 자동 생성함으로써 모바일 응용의 생산성을 향상시킨다.
2. 사용자 인터페이스로 생성된 카드 또는 데크의 표현 가능한 범위는 관심의 대상인 닷 엔티티 뿐만 아니라 닷 엔티티와 직·간접적으로 연관된 확장, 첨부, 확장-첨부 엔티티들까지 포함하기 때문에 데이터베이스 커버리지가 매우 높다.
3. MobileGen은 템플릿 기반으로 스크립트들을 생성하기 때문에 생성기 자체 및 생성된 응용의 버전 관리가 용이하다.

향후, MobileGen은 수요가 급증하는 모바일 GIS 응용의 자동 생성을 위해 맵 인터페이스 등의 기능을 추가해 모바일 GIS 응용 생성기로 발전시킬 계획이다.

참 고 문 헌

- [1] Mobile Device Technology and Trends, http://www.idc.com/getdoc.jsp?containerId=IDC_P1600, 2008.
- [2] B. Doyle and C. Lopes, Survey of Technologies for Web Application Development, ACM Press, 2006.
- [3] Jia Zhang et al, "Towards Increasing Web Application Productivity", Proceedings of the ACM symposium on Applied computing, March 14-17, Nicosia Cyprus, pp.1677-1681 2004.
- [4] MobileDev, http://www.speedwareltd.com/solutions/wireless/mobile_development_tools/MobileDev/, 2008.
- [5] WINDEV Mobile, <http://www.windev.com/windevmobile/index.html>, 2008.
- [6] MobiForms, <http://www.mobiforms.com/>, 2008.
- [7] Visual Studio.NET, <http://www.microsoft.com/korea/msdn/vstudio/>, 2008.
- [8] Doohun Eum and Toshimi Minoura, WebSiteGen: Web-Based Database Application Generator, IEICE Trans. on Information and Systems, Vol.E86-D, No.6, pp.1001-1010, June 2003.
- [9] Korry Douglas, PostgreSQL, 2nd Ed., Sams, 2005.
- [10] Paul DuBois, MySQL Cookbook, 2nd Ed., O'Reilly, 2007.
- [11] Amitava Biswas, et al, "Assessment of mobile experience engine, the development toolkit for context aware mobile applications", Proceedings of the ACM SIGCHI International Conference on Advances in Computer Entertainment Technology, June 14-16, Hollywood California, pp.1600-1613, 2006.
- [12] A. Biswas and J. Singh, "Software Engineering Challenges in New Media Applications", Proceedings of the Software Engineering Applications, November 13-15, Dallas Texas, pp.306-321, 2006.
- [13] Mary Flanagan, C. Daniel, Howe, and Helen Nissenbaum, "Values at play: design tradeoffs in socially-oriented game design", Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, April 2-7, Portland Oregon, pp.751-760, 2005.

● 저 자 소 개 ●



음 두 현

1984년 서강대학교 전자공학과 졸업(학사)
1987년 오레곤주립대학교 대학원 컴퓨터공학과 졸업(석사)
1990년 오레곤주립대학교 대학원 컴퓨터공학과 졸업(박사)
1992~현재 덕성여자대학교 컴퓨터공학부 교수
관심분야 : 객체지향 시스템
E-mail : dheum@duksung.ac.kr