KRISTAL IRMS 기반의 계층형 웹 게시판 개발

Development of Hierarchical Web Board Using KRISTAL IRMS

황미녕(Mi-Nyeong Hwang)*, 김진숙(JinSuk Kim)** , 윤화묵(HwaMook Yoon)***

In this paper, we introduce KBoard, a hierarchical web bulletin board based on KRISTAL IRMS. The KBoard has been developed in the form of general web bulletin board, and it supports various types of boards for general positing, downloads, and image gallery. Since the KBoard employs template files for web page design, the overall appearance of each board can be easily modified without changing the source code. The KBoard has an additional advantage that supports high-performance search in massive data environments because it is based on KRISTAL IRMS.

Keywords : KRISTAL IRMS, Board, Web

Ⅰ. 서론

KRISTAL IRMS(이하 KRISTAL)의 사용 분야가 많아지면서 일반 RDBMS를 사용하지 않고 KRISTAL만을 사용하여 웹 게시판으로의 활용 요구가 많아지게 되었다. 이런 요구에 맞추어 순수한 KRISTAL만을 이용하여 검색 기능이 지원되는 웹 게시판 응용프로그램을 개발하게 되었다. 본 논문에서는 KRISTAL 기반의 웹 게시판 KBoard에 대한 소개를 하고자 한다.

Ⅱ. KBoard (KRISTAL IRMS 기반 게시판)

1. 제로보드와 KBoard

1) 제로보드

제로보드는 PHP와 MySQL이 지원되는 서버에 설치하여 사용하는 공개 소스 게시판이다. 상업적인 목적이 아니라면 MySQL을 무료로 사용할 수 있고, PHP까지 지원이 된다면 제로보드를 이용해서 쉽게 게시판을 만들어 사용할 수 있기 때문에 국내에서는 제로보드를 지원해주느냐가 사용자가 서버 호스팅 업체를 선택하는 기준이 되고 있기 때문에 쉽게 제로보드를 사용할 수 있는 환경이 되고 있다. 제로보드는 1999년 배포 이후 여러 버전 업그레이드를 거쳐서 현재는 4.1 버전이 안정화되어 배포하고 있으며, 새로운 프레임웍을 적용한 버전 5도 베타 버전으로 배포되고 있다.

제로보드의 특징은 무료 게시판이면서도 다양한 기능을 제공한다. 이 다양한 기능의 제공에는 제로보드의 정책이 공개 소스를 지향한다는 것에서 기인하게 되었다. 소스를 공개했기 때문에 일부 적극적인 사용자들은 소스를 수정하여 좀 더 다양한 기능을 가진 게시판을 재배포하게 되었으며 이 활동이 활발해지면서 화면 디자인을

1)*한국과학기술정보연구원(Korea Institute of Science and Technology Information, mnhwang@kisti.re.kr),
**한국과학기술정보연구원(Korea Institute of Science and Technology Information, jinsuk@kisti.re.kr),
***한국과학기술정보연구원(Korea Institute of Science and Technology Information, hmyoon@kisti.re.kr)

뜻하는 스킨을 사용자들이 직접 수정하게 되었다. 그러나 스킨 변형에만 편중하여 추가적인 개발이 미비한 채로 시간을 끌어오다 올해 버전 5가 새롭게 선보이게 된 것이다. 이런 제로보드는 사용의 편리성에는 큰 점수를 줄 수 있으나 보안 문제가 가장 큰 약점으로 지목되고 있다. 사용자가 제로보드를 한 번 설치한 후에는 적극적으로 보안 패치를 하지 않는다는 점과 PHP의 유연한 특성으로 인한 보안 관련 문제와 일반 사용자들이 낮은 PHP에 적용된 제로보드를 사용하고 있기 때문에 서버 관리자들이 보안 업그레이드를 하기 어렵다는 문제 등 종합적으로 보안 관련하여 어려움을 보이고 있다. 이는 제로보드가 공개소스이면서 국내에서 제작되는 게시판의 상당수를 차지하고 있기 때문에 한 번 취약해진 보안성은 다수 사이트의 보안의 위험을 가져온다는 것에 있다.

그러나 쉽게 이 문제가 해결되지 않은 이유로는 제로보드의 소스 코드가 매우 혼잡하다는 것에서 기인된다. HTML 사용자 입출력, DB와의 연결 부분, 데이터 관리 부분 등 모든 코드가 섞여 있고, 이에 기반을 두어 각종 스킨들이 배포된 상황이여서 획기적인 수정이 힘들었다. 이 점을 보완하기 위해 버전 5 베타 버전이 배포되고 있지만 업그레이드에 적극적이지 않는 사용자들이 현재 사용하고 있는 제로보드를 얼마나 버전을 업그레이드 시킬지는 의문이다.

2) KBoard의 특징

KRISTAL IRMS 기반 게시판(이하 KBoard)은 제로보드의 기능적인 면을 참고하여 MySQL을 사용하지 않고 KRISTAL IRMS를 사용하여 만든 게시판이다. 현재 KRISTAL v3.1을 사용하고 있으며 KRISTAL을 이용한 검색 기능을 제공한다. 또한 계층형 게시판으로 만들어져 있으며 일반 사용자와 관리자 모드를 지원한다. 게시판 타입으로는 일반 게시판, 자료실 게시판, 갤러리 게시판이 있으며 첨부파일을 다운로드시 직접 링크를 사용하지 않는다.(제로보드는 직접 링크 사용) 또한 사용자의 입력을 받는 웹 인터페이스는 PHP로 작성하되 DB와 통신하는 실행 파일은 C++ 이진 파일이 그 역할을 담당하여 디자인을 관리하는 템플릿 파일이 HTML을 기초로 하여 따로 관리하도록 각 역할을 분리한 것이 큰 특징이다.

3) KBoard의 구성도

<그림 1> KBoard의 서비스 흐름도

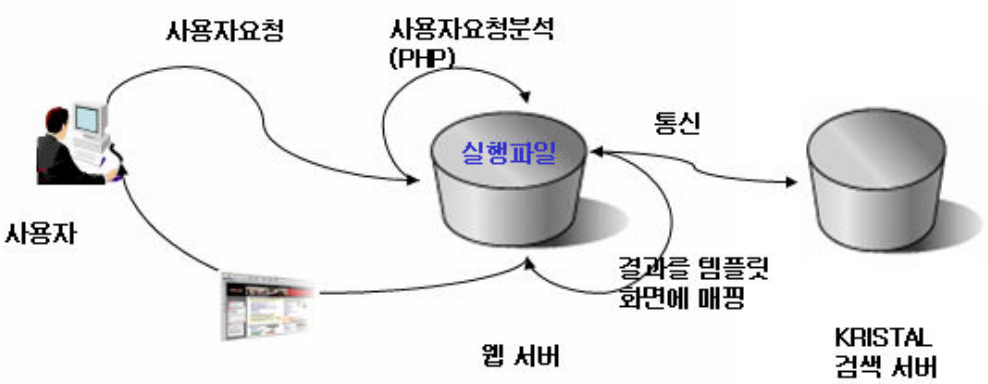


그림 1에서 보듯이 KBoard는 사용자가 요청을 하면 그 요청을 분석하여(PHP) KRISTAL 검색 서버와 통신을 하게 되고(C++ 이진 파일) 그 결과를 템플릿 파일을 이용하여(HTML) 화면에 매핑해서 사용자에게 보여주도록 구성되어 있다 .

4) 시스템 요구사항

<표 1> 시스템 요구 사항

KRISTAL 버전	KRISTAL 3.1
PHP	PHP 5이상, gd 라이브러리 설치
서버 환경	linux, g++
서버 구성	웹 서버와 검색 서버(DB)의 분리 가능

KBoard를 정상적으로 작동하여 사용하기 위해서 시스템 요구사항은 표 1과 같다. KRISTAL 버전은 3.1 이상, PHP 버전은 5이상으로 gd 라이브러리가 함께 설치되어야 한다. 웹 서버와 검색 서버를 분리하여도 무관하다.

5) 게시판 DB 스키마

KBoard의 게시판 DB 스키마는 아래와 같다. TITLE은 게시물의 제목, AUTHOR은 작성자 이름, PASSWD는 비밀번호, EMAIL은 이메일 주소, AUTHORIP는 작성자 IP 주소, CONTENT는 본문 내용, DATE는 작성 날짜(년월일시분초), UPFN1~UPFN5는 업로드한 파일이름으로 5개의 첨부파일을 허용한다. RUPFN1~RUPFN5는 업로드한 파일이 실제로 저장된 위치를 기록한다. 업로드한 파일은 파일명을 변환하여 암호화하여 저장하기 때문에 UPFN과 RUPFN는 그 정보가 서로 같지 않다. UPFNSZ1~UPFNSZ5는 업로드한 각 파일의 크기, DNFN1~DNFN5는 다운로드한 횟수를 담고 있다. LINK1, LINK2는 링크 정보, THREAD는 게시판을 계층형으로 정렬하기 위하여 KBoard에서 정의한 번호로 이후에 상세히 설명토록 하겠다. DEPTH는 이 글의 깊이를 표시한다. 0이면 새 글이고 1이상이면 답변 글이 된다. CATEGORY는 분류를 정의하고 NOTICE는 공지사항인지 여부를 REPLYMAIL은 회신 여부 체크, USEHTML은 HTML를 사용할 것인지 체크, ISSECRET은 비밀글 여부를 담는다. HIT는 조회 수 RECOMMEND는 추천 횟수, CHCOUNT는 현재 게시물이 아래로 몇 개의 답변 글을 가지고 있는지 정보를 담는다. 이는 THREAD, DEPTH와 어울려서 게시물을 계층형태로 보여주는데 필요하다. LEVEL은 작성자의 레벨, BOARDNAME은 게시판 이름이다.

<표 2> KBoard의 계층성

게시판 목록	번호	THREAD	DEPTH	CHCOUNT
오라클 DB로 부터 자동 데이터 수집하는 기능이 있었으면 좋겠습니다.	7	3000	0	2
↳ [re] 문헌정보서비스에서 크리스탈과 RDBMS의 역할 관계	6	2999	1	1
↳ [re][re] 문헌정보서비스에서 크리스탈과 RDBMS의 역할 관계	5	2998	2	1
↳ [re][re][re] 문헌정보서비스에서 크리스탈과 RDBMS의 역할 관계	4	2997	3	0
↳ [re] 논의를 시작하도록 해야겠습니다.	3	2996	1	0
할로윈	2	2000	0	0
데이터 입력시 processDocumentError 가 나는 이유가 뭔가요?	1	1000	0	0

```
<?xml version="1.0"?>
<DatabaseSchema>
        <CreateDatabase database-name="boardDB"   volume-dir="/data/home/acms/board_db/volume" />
                <CreateTableSchema name="boardDB_SCHEMA" use-aux-db="yes" doc-type="plain">
                <Stopword file=
                    "/data/home/acms/KRISTAL-3.1.1/examples3/stopword/swords-eng"/>
                <Stopword file=
                    "/data/home/acms/KRISTAL-3.1.1/examples3/stopword/swords-han"/>
                <!-- Record Index Section -->
                <BasicSection  name="TITLE"       data-type="KSTRING"      index-type="INDEX_BY_MA"/>
                <BasicSection  name="AUTHOR"   data-type="KSTRING"      index-type="INDEX_BY_TOKEN"/>
                <BasicSection  name="PASSWD"   data-type="KSTRING"      index-type="DO_NOT_INDEX"/>
                <BasicSection  name="EMAIL"       data-type="KSTRING"      index-type="INDEX_AS_IS"/>
                <BasicSection  name="AUTHORIP" data-type="KSTRING"      index-type="INDEX_AS_IS"/>
                <BasicSection  name="CONTENT"   data-type="KSTRING"      index-type="INDEX_BY_MA"/>
                <BasicSection  name="DATE"        data-type="KSTRING"      index-type="INDEX_AS_IS"/>
                <BasicSection  name="UPFN1"       data-type="KSTRING"      index-type="INDEX_BY_CHAR" DEFAULT-VALUE=""/>
                <BasicSection  name="UPFN2"       data-type="KSTRING"      index-type="INDEX_BY_CHAR" DEFAULT-VALUE=""/>
                <BasicSection  name="UPFN3"       data-type="KSTRING"      index-type="INDEX_BY_CHAR" DEFAULT-VALUE=""/>
                <BasicSection  name="UPFN4"       data-type="KSTRING"      index-type="INDEX_BY_CHAR" DEFAULT-VALUE=""/>
                <BasicSection  name="UPFN5"       data-type="KSTRING"      index-type="INDEX_BY_CHAR" DEFAULT-VALUE=""/>
                <BasicSection  name="RUPFN1"      data-type="KSTRING"      index-type="DO_NOT_INDEX" DEFAULT-VALUE=""/>
                <BasicSection  name="RUPFN2"      data-type="KSTRING"      index-type="DO_NOT_INDEX" DEFAULT-VALUE=""/>
                <BasicSection  name="RUPFN3"      data-type="KSTRING"      index-type="DO_NOT_INDEX" DEFAULT-VALUE=""/>
                <BasicSection  name="RUPFN4"      data-type="KSTRING"      index-type="DO_NOT_INDEX" DEFAULT-VALUE=""/>
                <BasicSection  name="RUPFN5"     data-type="KSTRING"     index-type="DO_NOT_INDEX" DEFAULT-VALUE=""/>
                <BasicSection  name="UPFNSZ1"     data-type="KINT"         index-type="DO_NOT_INDEX" DEFAULT-VALUE="0"/>
                <BasicSection  name="UPFNSZ2"     data-type="KINT"         index-type="DO_NOT_INDEX" DEFAULT-VALUE="0"/>
                <BasicSection  name="UPFNSZ3"     data-type="KINT"         index-type="DO_NOT_INDEX" DEFAULT-VALUE="0"/>
                <BasicSection  name="UPFNSZ4"     data-type="KINT"         index-type="DO_NOT_INDEX" DEFAULT-VALUE="0"/>
                <BasicSection  name="UPFNSZ5"     data-type="KINT"         index-type="DO_NOT_INDEX" DEFAULT-VALUE="0"/>
                <BasicSection  name="DNFN1"       data-type="KINT"         index-type="DO_NOT_INDEX" DEFAULT-VALUE="0"/>
                <BasicSection  name="DNFN2"       data-type="KINT"         index-type="DO_NOT_INDEX" DEFAULT-VALUE="0"/>
                <BasicSection  name="DNFN3"       data-type="KINT"         index-type="DO_NOT_INDEX" DEFAULT-VALUE="0"/>
                <BasicSection  name="DNFN4"       data-type="KINT"         index-type="DO_NOT_INDEX" DEFAULT-VALUE="0"/>
                <BasicSection  name="DNFN5"       data-type="KINT"         index-type="DO_NOT_INDEX" DEFAULT-VALUE="0"/>
                <BasicSection  name="LINK1"      data-type="KSTRING"      index-type="INDEX_AS_IS" DEFAULT-VALUE=""/>
                <BasicSection  name="LINK2"      data-type="KSTRING"      index-type="INDEX_AS_IS" DEFAULT-VALUE=""/>
                <BasicSection  name="THREAD"    data-type="KINT"         index-type="INDEX_AS_NUMERIC"/>
                <BasicSection  name="DEPTH"     data-type="KINT"         index-type="INDEX_AS_NUMERIC" DEFAULT-VALUE="0"/
                <BasicSection  name="CATEGORY" data-type="KSTRING"     index-type="INDEX_AS_IS" DEFAULT-VALUE=""/>
                <BasicSection  name="NOTICE"     data-type="KBOOL"       index-type="INDEX_AS_IS" DEFAULT-VALUE="FALSE"/>
                <BasicSection  name="REPLYMAIL" data-type="KBOOL"    index-type="DO_NOT_INDEX" DEFAULT-VALUE="FALSE"/>
                <BasicSection  name="USEHTML"    data-type="KBOOL"    index-type="DO_NOT_INDEX" DEFAULT-VALUE="FALSE"/>
                <BasicSection  name="ISSECRET"   data-type="KBOOL"    index-type="DO_NOT_INDEX" DEFAULT-VALUE="FALSE"/>
                <BasicSection  name="HIT"           data-type="KINT"       index-type="INDEX_AS_NUMERIC" DEFAULT-VALUE="0"/>
                <BasicSection  name="RECOMMEND" data-type="KINT"     index-type="INDEX_AS_NUMERIC" DEFAULT-VALUE="0"/>
                <BasicSection  name="CHCOUNT"    data-type="KINT"     index-type="INDEX_AS_NUMERIC" DEFAULT-VALUE="0"/>
                <BasicSection  name="LEVEL"      data-type="KINT"     index-type="INDEX_AS_NUMERIC" DEFAULT-VALUE="9"/>
                <BasicSection  name="BOARDNAME" data-type="KSTRING"    index-type="INDEX_AS_IS"/>
 <!-- Virtual Section Definitions -->
            <!-- Union Section Definitions -->
                <UnionSection name="TITCON"       include-sections="TITLE CONTENT"/>
                <UnionSection name="TITCONAU"    include-sections="AUTHOR TITLE CONTENT"/>
                <UnionSection name="ALL"          include-sections="TITLE AUTHOR EMAIL CONTENT LINK1 LINK2" />
                </CreateTableSchema>
        <CreateTable table-name="boardDB_TABLE" with-schema="boardDB_SCHEMA"/>
</DatabaseSchema>
```

표 2에서 보듯이 새로운 글이 작성되면 THREAD값은 가장 큰 THREAD값에서 1000을 더하여 그 값을 부여하고 DEPTH와 CHCOUNT는 자동적으로 0이 들어간다. 답변 글의 경우에는 THREAD는 자신의 부모 글의 THREAD-1이 되고, DEPTH는 부모+1, CHCOUNT는 0이 된다. 그러면서 자신의 부모 글의 CHCOUNT+1 해주게 된다. 이렇게 하는 이유는 자식 글 즉 답변 글이 있는 경우는 부모 글을 삭제할 수 없도록 하기 위해서이다. (단, 관리자의 경우에는 게시판 목록에서 다수 삭제가 가능한 권한을 따로 부여한다.) 이와 동시에 새로 들어간 자식 글 THREAD값을 하나씩 감소시킨다. 표 2의 게시판 목록에서 글이 작성된 순서는 1->2->7->3->6->5->4 이다. 따라서 번호 7번 아래의 첫 번째 답변글은 3번 글로 3번 글이 입력되는 당시에는 3번 글의 THREAD는 2999가 되고 DEPTH는 1, CHCOUNT는 0이 된다. 그러나 6번 글이 다시 7번 글의 답변글로 달리면서 3번 글은 THREAD는 2998로 감소되고, 6번 글이 2999번이 된다. 이어 6번 글의 답변글로 5번 글이 달리면서 5번 글이

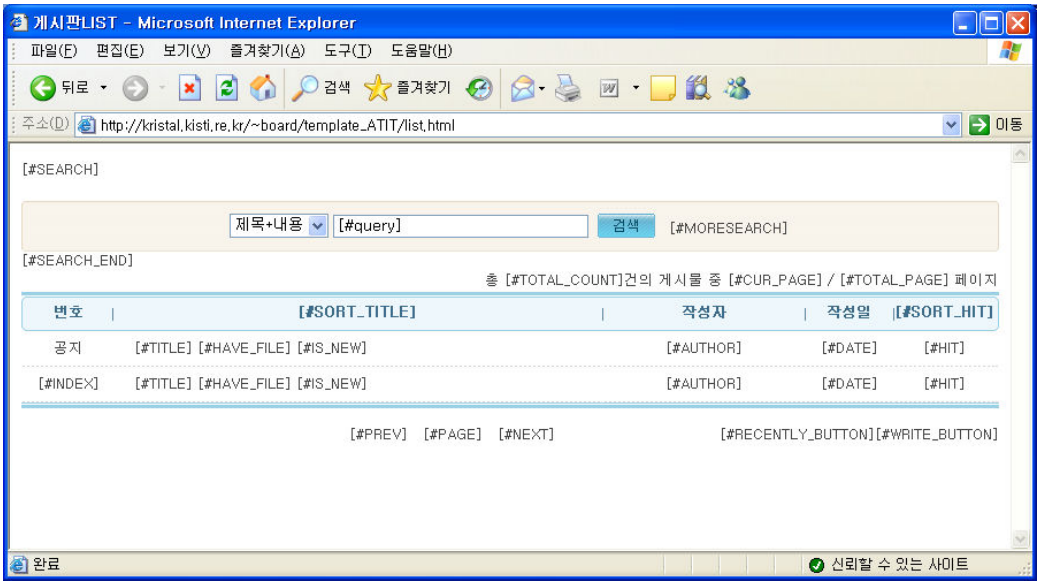
2998이 되고, 3번 글은 2997로 1이 감소된다. 이후 5번 글에 4번 글이 답변 글이 되면서 다시 3번 글이 2996이 되게 된 것이다.

이런 방식을 부분 업데이트형이라고 부를 수 있는데 신규 글이 등록되면 THREAD값들의 업데이트는 없지만 답변 글이 입력되면 그 이하의 답변 글이 있는 경우에만 THREAD값이 변경되기 때문에 DB의 수정 빈도가 낮은 편이다. 이렇게 THREAD 정책을 사용하게 되면 THREAD값을 기준으로 내림차순으로 정렬하면서 DEPTH값을 기준으로 들여쓰기를 통해서 게시판의 계층형태 목록을 가시화할 수 있어서 편리한 장점이 있는 있다. 반면에 신규 글만 작성될 때에는 1000씩 THREAD값이 늘어나 THREAD값이 과장되게 사용된다는 약점도 있다. 글의 삭제는 CHCOUNT가 0인 글, 즉 답변 글(자식글)이 없는 경우만 허용한다. 즉 표 2에서는 글 번호 7, 6, 5는 삭제가 불가능하다. 이런 삭제 정책은 제로보드에서도 동일하게 사용하고 있다. 관리자 권한일 때에는 목록 보기에서 다중 선택을 동해 일괄 삭제 기능을 지원해주고 있다.

6) 템플릿 파일

KBoard에서는 웹 브라우저에 뿌려지는 화면 디자인을 소스 코드에서 삽입하지 않고 템플릿 파일로 따로 관리하고 있다. 이는 기본적은 HTML에 KBoard에서 정의한 일부 예약어를 삽입함으로 해서 KBoard의 파서가 템플릿 파일을 읽어 들여 검색 서버에서 얻은 결과를 해당하는 예약어와 치환하여 HTML를 완성한 후 화면에 보여주는 것으로 정의된 규칙만 이해한다면 자유롭게 화면 디자인 변경이 가능하다.

<그림 2> 템플릿 화면을 웹 브라우저로 봤을 때



템플릿 파일은 “목록보기”, “상세보기”, “입력/수정”, “상세보기” 등 4개의 파일로 나뉘어져 있다. 이 목록보기의 템플릿 파일의 형태에 따라 화면이 일반 게시판 목록으로 보이기도 하고 갤러리 게시판의 형태로 보이기도 한다. 그림 2는 목록보기 템플릿 파일을 웹 브라우저로 보여주고 있다. HTML에

부분적으로 예약어를 넣은 것으로 특정한 규칙에 의해 예약어가 정의되어 있다. [#TITLE]은 제목 [#HAVE_FILE]은 첨부파일이 있는 경우 보이는 아이콘, [#IS_NEW]는 최신 글 표시를 의미한다. 첨부파일이 있는 경우 화면에 표시되는 아이콘 역시 템플릿 파일의 상단에서 이미지를 지정, 수정할 수 있다. [#AUTHOR]는 작성자, [#DATE]는 날짜, [#HIT]는 조회 수가 예약어와 치환되어서 화면에 출력된다. 템플릿 파일과 관련된 자세한 내용은 KBoard 매뉴얼을 참고하면 된다.

3. KBoard 서비스 이용

KBoard는 공개 소프트웨어이기 때문에 누구라도 이용이 가능하다. http://www.kristalinfo.com에서 무료 배포하고 있다. 설치방법과 관련 매뉴얼이 링크되어 있으므로 직접 설치해볼 수 있다. KBoard를 설치한 후 중요하게 볼 파일이 2개가 있는데 boardlist.txt 파일에는 게시판 생성, 카테고리 정의, 글을 작성할 수 있는 권한 설정, 게시판 글 쓰는데 HTML 태그 허용여부 결정, 업로드 데이터 용량 결정 등등이 있고 admin.txt에는 관리자 관련 정보가 들어있고, 관리자가 관리하는 게시판 리스트 등을 정의할 수 있다. KBoard의 다음 버전에서는 파일로 관리하는 이 2가지 정보를 DB화할 계획이다.

Ⅲ. 결론

KBoard는 KRISTAL IRMS를 사용하여 만든 계층형 웹 게시판으로 게시물의 등록 즉시 검색이 가능하여 대용량 게시물에서도 효과적인 검색 결과를 보여준다. 현재 KBoard는 학회마을 106개의 학회에 게시판으로 사용되고 있으며 http://www.kristalinfo.com에서도 게시판으로 사용 중이다. 1차 버전에서의 미진했던 부분을 보완하여 2차 버전을 개발 중이며 2차 버전은 2007년도에 공개 예정이다.

[참고문헌] (바탕 10, 진하게)

http://zeroboard.com
http://www.kristalinfo.com
http://society.kisti.re.kr