Journal of Natural Sciences
Pai Chai University, Korea
Vol. 18, No. 1 : 17-28, 2007

Adobe Connect 콘텐츠 검색 플래시 어플리케이션 개발

박종대 · 조창호
배재대학교 전산전자물리학과 (우) 302-735 대전시 서구 도마동 439-6
jdpark@pcu.ac.kr, cho51111@pcu.ac.kr

Development of A Flash Application for Adobe Connect Contents Search

Jong-Dae Park and Chang-Ho Cho
Department of Computational Electronic Physics, Pai Chai University

요 약

Adobe사의 Connect 서버의 XML 웹서비스를 사용하여 콘텐츠를 검색할 수 있는 플래시 어플리케이션을 개발하였다. 개발된 어플리케이션을 통하여 서버주소를 알 수 있는 기관의 공개 콘텐츠를 검색할 수 있었다.

Abstract

We have developed a Adobe Connect contents search flash applications using XML web services. Contents search feasibility from the known Adobe Connect servers was demonstrated.

Corresponding Author : Jong-Dae Park, Department of Physics, Pai Chai University, Daejeon, Korea, 302-735, Tel : +82-42-520-5374, E-mail : jdpark@pcu.ac.kr

I. 서 론

Adobe Connect Enterprize 서버 제품은[1] Connect Presenter와 Connect Meeting, Connect Training 등 세 가지로 구성되어 있다. Connect Presenter는 파워포인트에 플러그인을 설치하여 사용하며, 파워포인트 강의 자료를 슬라이드별로 녹음하고 편집하여 이를 플래시 형태의 동영상 강의로 만들어 주는 솔루션이다. 녹음된 플래시 동영상 강의는 CD롬이나 PC에 저장하거나 Adobe Connect Enterprize 서버에 저장할 수 있다. 서버에 저장된 콘텐츠들은 각각 고유의 URL 주소를 가지게 되며 이를 Moodle등의 LMS등에서 링크하여 사용할 수 있다. Adobe Connect Meeting은 실시간 화상 강의를 할 수 있는 솔루션이며, 이를 녹화하여 추후 수업자료로 활용할 수 있다.

Adobe Connect Enterprize 서버 사용자들은 서버에 로그인 한 후 검색창을 통하여 콘텐츠들을 검색하여 공개된 동영상의 경우 수업자료로 활용할 수 있다. 그러나 서버에 등록되지 않은 사용자들은 서버에 공유된 콘텐츠들을 보는데 어려움이 있다. Adobe Connect Enterprize 서버는 다른 시스템과의 통합을 위하여 웹서비스를 xml 형태로 제공하는데 이를 이용하면 서버에서 많은 정보를 추출해 낼 수 있다.

본 논문에서는 Connect Enterprize 서버의 웹 서비스 기능[2]을 이용하여 서버 내의 콘텐츠를 검색할 수 있는 방법을 논의하고, 이를 구현한 플래시 어플리케이션을 소개하고자 한다.

II. Adobe Coonect 서버 웹서비스 소개

배재대학교가 사이트 라이센스를 받아 운영하고 있는 Adobe Connect 서버 주소는 http://br.pcu.ac.kr 이다. 이 서버는 250명까지 사용자가 가능하며, 계정을 만들기 위해서는 배재대학교 이러닝 연구소에 문의하면 된다. 이 서버 사용자는 다음 그림의 오른쪽 상단에 보이는 콘텐츠 검색창을 통하여 콘텐츠를 검색할 수 있다.

그림 1. Adobe Connect 화면

다음은 검색창에 "law" 라는 검색어를 입력한 후 검색한 결과들이다.

그림 2. Adobe Connect 콘텐츠 검색 화면

이러한 콘텐츠 검색은 Adobe Connect 서버의 웹서비스를 호출함에 의해서도 가능하다. Adobe Connect 6.0이전 버전인 Breeze 4.0, 5.0 버전에서는 서버에 로그인 함이 없이 콘텐츠 검색이 가능하였는데, Adobe Connect 6.0 버전 부터는 웹서비스로 로그인을 하여야 콘텐츠 검색이 가능하다.

웹 서비스로 로그인한 후 서버와의 지속적인 연결은 쿠키를 통해서 하게 된다. 쿠키는 웹페이지를 방문함에 의해서 만들어 진다. 다음은 인터넷 브라우저의 주소줄에

http://Adobe Connect 서버 주소/api/xml?action=common-info를 입력하면 받아오게 되는 xml 이다. 배재대학교의 경우에 Adobe Connect 서버주소는 http://br.pcu.ac.kr이므로 http://br. pcu.ac.kr/api/xml?action=common-info를 입력한다. 이 xml에는 웹서비스 요청한 것에 대하여 정상 처리되었음을 보여주는 status code가 "ok" 임을 알 수 있고, 쿠키가 어떤 값인지를 알 수 있다.

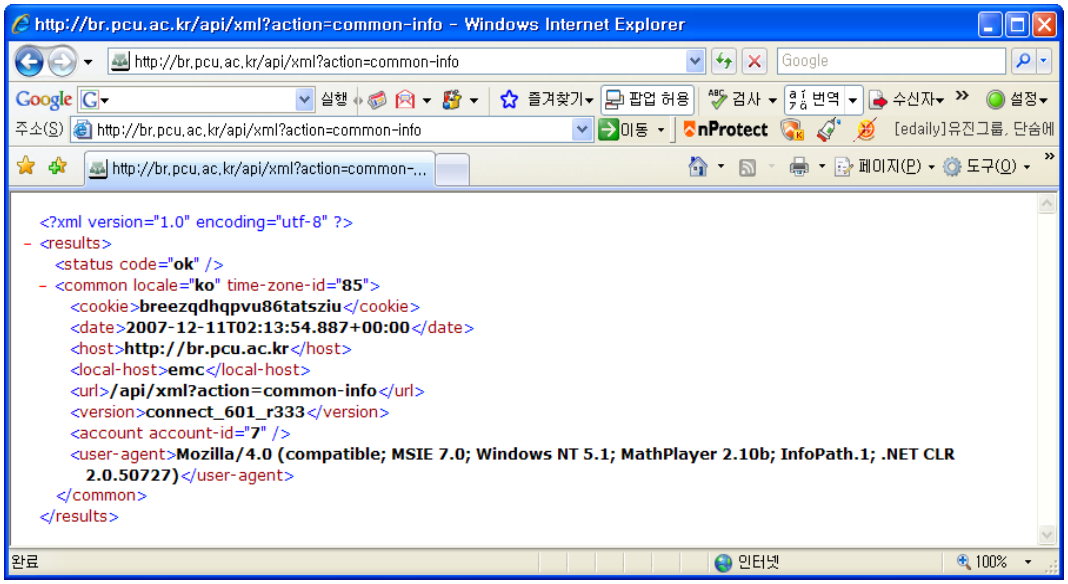

그림 3. Adobe Connect 웹서비스 활용 기본 정보 조회 결과–배재대학교

미국 인디아나 주립대학의 Adobe Connect 서버의 정보를 웹서비스로 가져오면 다음과 같다.

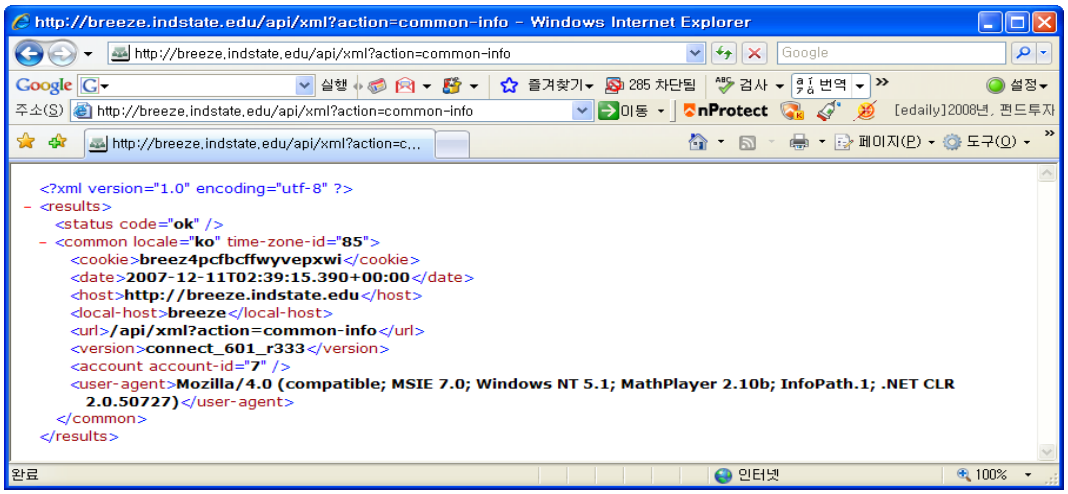

그림 4. Adobe Connect 웹서비스 활용 기본 정보 조회 결과–인디아나 대학교

인디아나 대학교에서도 배재대학교와 같은 버전의 서버를 사용하고 있음을 알 수 있다.

인디아나 대학의 Adobe Connect 서버에서 키워드가 "law"로 되어 있는 콘텐츠를 검색하면 다음과 같다.

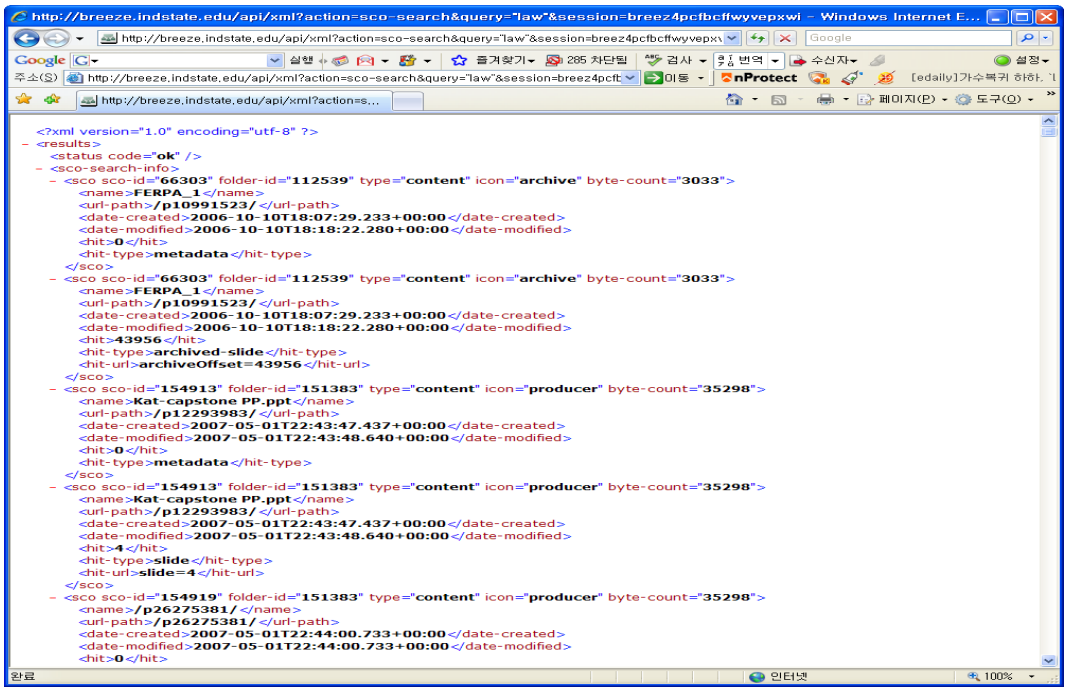

그림 5. Adobe Connect 웹서비스 활용 검색 결과–인디아나 대학교

이와 같이 공개된 콘텐츠들은 웹서비스를 사용하여 콘텐츠에 대한 정보를 볼 수 있으므로 이를 활용하면 다른 교육기관에서 운영 중인 Adobe Connect 서버의 콘텐츠 정보들을 볼 수 있다.

III. 플래시 어플리케이션 제작

Adobe Connect 서버의 웹서비스기능을 이용하여 서버에 저장되어 있는 콘텐츠를 검색하기 위한 플래시 어플리케이션을 제작하기 위한 도구로는 Macromedia Flash MX Professional 2004 [3]를 사용하였다.

여러 교육기관에서 공개된 Adobe Connect 콘텐츠를 검색하기 위해 교육기관을 목록에서 선택할 수 있는 콤보박스를 구성요소 창에서 드래그해서 장면으로 가져다 놓고 객체이름을 univ_cb 로 설정한다. 검색어 입력을 위한 텍스트 입력창을 마찬가지 방법으로 만들고 객체이름을 search_ti 로 설정한다. 검색된 콘텐츠의 수는 result_ta 텍스트 영역에 나타나도록 한다. 검색을 위한 버튼이름은 search_btn으로 한다. 검색 결과는 데이터그리드를 사용하며 콘텐츠 제목, 조회수, 링크들이 나타날 수 있도록 한다. 구성된 플래시 어플리케이션의 사용자 인터페이스는 그림 6과 같다.

그림 6. 플래시 어플리케이션 구성

그림 6에서 구성한 각 컴포넌트들이 유기적으로 동작하기 위해서는 플래시에 액션스크립트를 작성해 주어야 한다.

우선 사용자는 검색어 창에 검색어를 입력하고 search 버튼을 클릭하게 되는데 이때 search 버튼에 입력한 액션 스크립트는 다음과 같다.

버튼을 클릭하면 쿼리문을 xml형식으로 팩키지하여 Adobe Connect 서버로 보내게 되며, 응답 xml은 yourXML에 저장되게 된다. yourXML이 로드되면 init 함수가 호출되어 데이터그리드의 내용을 갱신하게 된다.

```
1  on(click){
2                  yourXML.xmlDecl="";
3                  this._parent.myDg_dg.removeAll();
4                  var scoSearch:String=this._parent.Search_ti.text;
5                  if(scoSearch !="")
6                           {
7                           var myString:String="<params>"
8                           +"<param name=\"action\">sco-search</param>"
9                           +"<param name=\"query\">"
10                          +scoSearch
11                          +"</param>"
12                          +"<param name=\"session\">"
13                          +sessNum
14                          +"</param>"
15                          +"</params>";
16                          myXML=new XML(myString);
17                          myXML.xmlDecl="<?xml version=\"1.0\"?>";
18                          var headers = ["Content-Type", "text/xml"];
19                          myXML.addRequestHeader(headers);
20                          var urls:String=this._parent.urlstr+"/api/xml";
21                          myXML.sendAndLoad(urls,yourXML);
22                          }
23 }
```

다음은 플래시 프로그램의 첫 번째 프레임에 입력한 액션스크립트 리스트이다. 프로그램의 1-4 번째 줄에서 데이터그리드의 열을 "contents", "hit", "link"로 나누고 각각의 폭을

지정하였고 9-22번째 줄에서는 Adobe Connect 를 사용하고 있는 교육기관의 서버 주소 정보를 입력하여 사용자가 콤보박스에서 선택할 수 있도록 하였다. 만일 Adobe Connect 서버 주소를 알 수 있다면 정보를 추가하여 검색할 수 있다.

```
1 myDg_dg.columnNames=["contents","hit","link"];
2         myDg_dg.getColumnAt(0).width=200;
3         myDg_dg.getColumnAt(1).width=150;
4         myDg_dg.getColumnAt(2).width=200;
5         var univ_cb:mx.controls.ComboBox;

6         _global.sessXML=new XML();
7         _global.yourXML=new XML();
8         _global.urlstr="http://br.pcu.ac.kr";
9         univ_cb.addItem("배재대학교","http://br.pcu.ac.kr");
10        univ_cb.addItem("인디아나주립대학","http://breeze.indstate.edu");
11        univ_cb.addItem("퍼듀대학","http://breeze-qa.test.ics.purdue.edu");
12        univ_cb.addItem("노스다코타 대학교","http://cilt.breeze.und.edu:8080");
13        univ_cb.addItem("국립싱가폴대학교","http://137.132.1.184");
14        univ_cb.addItem("미네소타대학교","http://breeze4.umn.edu");
15        univ_cb.addItem("클렘슨대학교","http://breeze.clemson.edu");
16        univ_cb.addItem("영국중부랭카셔대","http://breeze01.uclan.ac.uk");
17        univ_cb.addItem("캐나다 캘거리대","http://breeze.ucalgary.ca");
18        univ_cb.addItem("윈스콘신대","http://breezepresentation.uwstout.edu");
19        univ_cb.addItem("워싱톤대,세이트루이스","http://train2web.wustl.edu");
20        univ_cb.addItem("남부유타대","http://breezep.suu.edu");
21        univ_cb.addItem("마샤추세츠대,보스톤","http://media.umb.edu");
22        univ_cb.addItem("말레이지아툰압둘라작대","http://breeze.unitarklj1.edu.my");

23        cblistener = new Object();

24        var seString:String="<params>"
25        +"<param name=\"action\">common-info</param>"
26        +"</params>";
27        seXML=new XML(seString);
```

```
28        seXML.xmlDecl="<?xml version=\"1.0\"?>";
29        var headers = ["Content-Type", "text/xml"];
30        seXML.addRequestHeader(headers);
31        urlstr=univ_cb.selectedItem.data;
32        var urls:String=urlstr+"/api/xml";
33        seXML.sendAndLoad(urls,sessXML);

34        univ_cb.addEventListener("change", cblistener);
35        var oChangeEvent:Object=new Object();
36        oChangeEvent.type="change";
37        oChangeEvent.target="cblistener";
38        univ_cb.dispatchEvent(oChangeEvent);

39        cblistener.change = function(eventObject){
40        urlstr=univ_cb.selectedItem.data;
41        var urls:String=urlstr+"/api/xml";

42        seXML.sendAndLoad(urls,sessXML);
43        }

44        function resp(){

45        var snScoNode:XMLNode=sessXML.firstChild.nextSibling.firstChild.nextSibling;
46        _global.sessNum=snScoNode.firstChild.firstChild;

47        }
48        sessXML.onLoad=resp;

49 function init(){

50
var xnScoNode:XMLNode=yourXML.firstChild.nextSibling.firstChild.nextSibling.firstChild;
51
var scoArr:Array=yourXML.firstChild.nextSibling.firstChild.nextSibling.childNodes;
```

```
52                  result_ta.text=scoArr.length;
53                  for(var i:Number=0;i<scoArr.length;i++){
54                          var xnScoNodeName:XMLNode=scoArr[i].firstChild;
55                          var xnScoNodeURL:XMLNode=xnScoNodeName.nextSibling;
56                          var xnScoNodeDateCr:XMLNode=xnScoNodeURL.nextSibling;
57                          var xnScoNodeDateMo:XMLNode=xnScoNodeDateCr.nextSibling;
58                          var xnScoNodeAuthor:XMLNode=xnScoNodeDateMo.nextSibling;
59                          var oSco:Object=new Object();
60                          oSco.contents=xnScoNodeName.firstChild;
61                          oSco.link=urlstr+xnScoNodeURL.firstChild;
62                          oSco.author=xnScoNodeAuthor.firstChild;
63                          myDg_dg.addItem(oSco);
64                  }
65          }
66          yourXML.onLoad=init;

67          var myListener = new Object();
68          myListener.cellPress = function(event) {
69                  var cell:Number = event.itemIndex;

70
var xnScoNode:XMLNode=yourXML.firstChild.nextSibling.firstChild.nextSibling.firstChild;
71
var scoArr:Array=yourXML.firstChild.nextSibling.firstChild.nextSibling.childNodes;
72                          var xnScoNodeName:XMLNode=scoArr[cell].firstChild;
73                          var xnScoNodeURL:XMLNode=xnScoNodeName.nextSibling;
74                          var slink=urlstr+xnScoNodeURL.firstChild;
75                          getURL(slink,"_blank");
76          };
77          myDg_dg.addEventListener("cellPress", myListener);
```

다음은 앞에서 설명한 것과 같은 방법으로 개발한 플래시 어플리케이션의 동작 모습이다. 인디아나 주립대학의 Adobe Connect 서버에서 "law" 를 검색어로 입력하여 얻은 결과

를 보여주고 있다. 총 262건이 검색되었음을 알 수 있다.

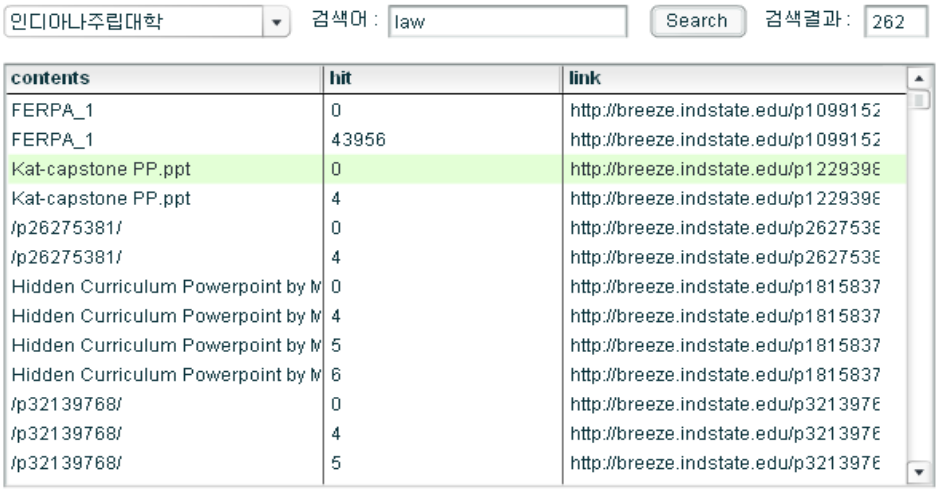

그림 7. 콘텐츠 검색 결과

Adobe Connect 5.0 이전 버전에서는 콘텐츠의 저자 정보가 함께 제공되었으나 6.0 부터는 저자 정보가 제공되고 있지 않다. 웹서비스에서 콘텐츠에 대한 간략한 설명도 제공한다면 사용자들이 콘텐츠를 찾아보기가 쉬울 것이다. Adobe Connect 6.0의 차기 버전인 Brio[4]에서 이러한 기능 향상을 기대해 본다.

IV. 결 론

웹서비스는 SOA(Service Oriented Architecture) 에서 중요한 역할을 한다. 웹서비스를 활용하여 Loosely Coupled System을 구축할 수 있다. 본 논문에서는 Adobe Connect 서버의 웹 서비스를 사용하여 콘텐츠 정보를 검색할 수 있는 플래시 어플리케이션 개발에 대해 논의하였다. 저작도구로는 플래시 MX 2004 [3,4]를 사용하였으며, 검색된 결과를 바로 링크를 클릭하여 콘텐츠를 볼 수 있도록 하였다. 이와 같은 콘텐츠 검색기는 콘텐츠 공유를 용이하게 한다.

참고문헌

1. http://br.pcu.ac.kr
2. http://help.adobe.com/en_US/Connect/6.0/WebServices/help.pdf
3. http://www.adobe.com/products/flash/
4. http://labs.adobe.com/technologies/brio/