AUTOMATED CONFERENCE CD-ROM BUILDER – AN OPEN SOURCE APPROACH

Stefan Karastanev

***Abstract**: This paper presents a new approach for creating conferences CD based HTML presentation of published papers for off-line reading. A couple of advances techniques has been deployed to automate process of creation like Java script engine, SQL database integration, full text search engine. The approach offers a good integration with online conference management systems for direct data exchange, friendly user interface and good compatibility with majority of contemporary HTML browsers.*

***Key Words**: Java script, SQL, PHP, Database, Open Source, Linux.*

Introduction

Creation of presentation CD for offline publishing is usually a complex task which has to solve several problems. First of them is to be as much as possible versatile in terms of machine platforms and operating systems. Second of them is to avoid installation of any specialized software components on target machine. Of course the speed of operation is important as well. This leads to deployment of HTML based techniques for reading, searching and browsing the published content, which ensures full platform and OS independence, light and convenient user interface and finally avoids any concerns about security issues.

There are known commercial conference CD builders, but they have number of disadvantages like proprietary license, lack of possibilities for integration with well known online conference management systems, necessity of availability of installed third party engines (Java Virtual Machine for example) or executable modules, which may compromise the security of the target machine.

Goals and tasks

The main goals, according to the introduction above, are to develop robust HTML based CD conference publishing system for offline reading with clear user interface, to ensure needed functionality for reading, browsing and searching the content without use of any external virtual machines or database engines, leaning only on HTML browser 's internal capabilities. The system should be based entirely on open source projects, libraries or **application programming interfaces** (API).

Research methods and models

Taking into account the constrains of the HTML and the need of generation of dynamic content there are two methods to apply to achieve the goals highlighted above – deploying of build-in Java Scrip engine available in most of popular browsers or usage of third party script engines (e.g. Macromedia Flash) which requires presence of additional plugin installed on the user's machine. In this paper the Java Scrip approach has been chosen.

There are known number of Java Script frameworks licensed under open source licenses like Dojo Toolkit [1], MooTools [2] and many others but probably most convenient and with rich functionality framework for building Javascript applications is **qooxdoo** [3]. Qooxdoo is a comprehensive and innovative framework for creating rich internet applications (RIAs). Leveraging object-oriented JavaScript allows developers to build impressive cross-browser applications. No HTML, CSS nor DOM knowledge is needed. It includes a platform-independent development tool chain, a state-of-the-art GUI toolkit and an advanced client-server communication layer. It is open source under an LGPL/EPL dual license. Despite being a pure JavaScript framework, qooxdoo is quite on par with GUI toolkits like Qt or SWT when it comes to advanced yet easy to implement user interfaces. It offers a full-blown set of widgets that are hardly distinguishable from elements of native desktop applications. Full built-in support for keyboard navigation, focus and tab handling and drag & drop is provided. Dimensions can be specified as static, auto-sizing, stretching, percentage, weighted flex or min/max or even as combination of those. All widgets are based on powerful and flexible layout managers which are a key to many of the advanced layout capabilities. Interface description is done programmatically in JavaScript for maximum performance.

No HTML has to be used and augmented to define the interface. The qooxdoo developer does not even have to know CSS to style the interface. Clean and easy-to-configure themes for appearance, colors, borders, fonts and icons allow for a full-fledged styling that even supports runtime switching.

Picking up a future rich Java Script framework solving the basic program organization and user interface tasks. The next step of the application development is to choose appropriate SQL database engine. Why SQL database? Using such approach separates code from the data which gives flexibility in usage of the application in wide variety of data sources (online conference systems, manual data entering etc.). TrimPath Query [4] has been chosen as Javascript SQL engine. This engine provides simple SQL interface with json like database storage.

The next step is to choose and integrate a full text search engine. JSSIndex java script engine has been chosen [5]. JSS is a simple search engine designed for CDROM or Web-based document collections. The documents to be indexed can be in HTML, PostScript (.ps and .ps.gz), PDF, and DjVu. The main feature of JSS is that the query engine and the index are entirely in JavaScript, and therefore require no other software than a JavaScript-enabled Web browser. JSS uses a simple inverted word list. Each individual word that appears in the corpus of documents is associated with a list of documents (or a list of pages) in which the word appears.

The inverted word list is encoded in string literals in JavaScript source code. With this trick, the search engine and the index are entirely contained in a piece of JavaScript source code that runs in the browser. For CDROM-based collections, this provides a machine-independent search capability without requiring any software installation, and without requiring a Java virtual machine. For web-based collections, this provides a simple search capability, without requiring any server-side software installation, and without consuming any server resources. The indexing piece of code is written in Lush (a high level script language). Some small modifications has been done to improve PDF to text conversion, which has been executed on Linux machine during compilation of the published papers data.

The simplified model of the application is shown at the figure 1. All components reside in the browser's Javascript engine environment.

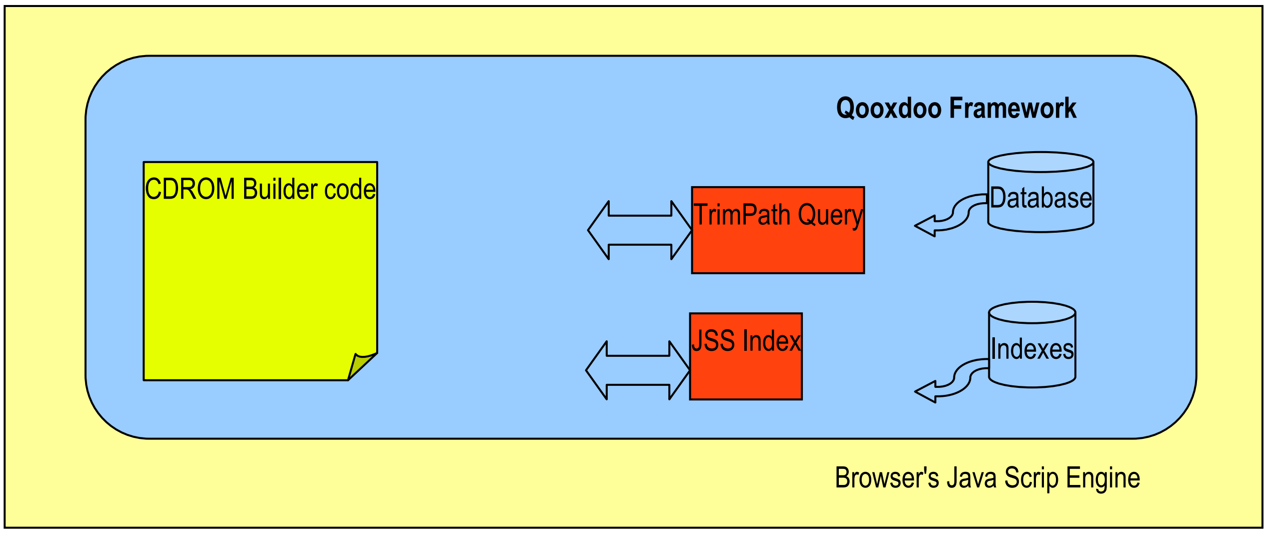


Fig.1 Application Model

- **Realizations and experiments**

The realization described in this paper is built on top of data of 11-th National Congress on Theoretical and Applied Mechanics. According to the application model, the main code is written in Java Script, which is processed by qooxdoo's framework build script to produce final optimized and compressed script. An additional proxy html file is used to load main java script file and auxiliary scripts for sql and full text search engines. The main script consists of the following important blocks:

- Main window block – initializes the qooxdoo framework, creates two containers – menu area and information area (figure 2)

- Menus block – creates the necessary menu items as **Tree List** and installs an event listener to the **Change Selection** event from the **Tree List** item. Most of the items have a static realization because they don't depend of the concrete conference but the **Section** item is specific for every realization so it's filled by the code with help of the sql queries. The code looks like the following segment :

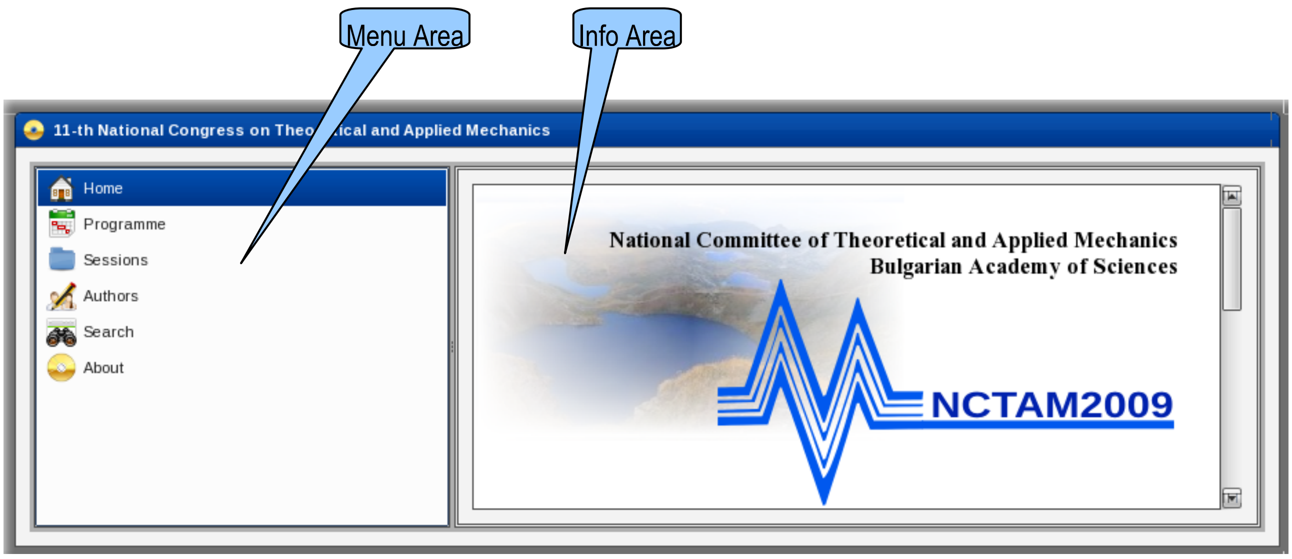
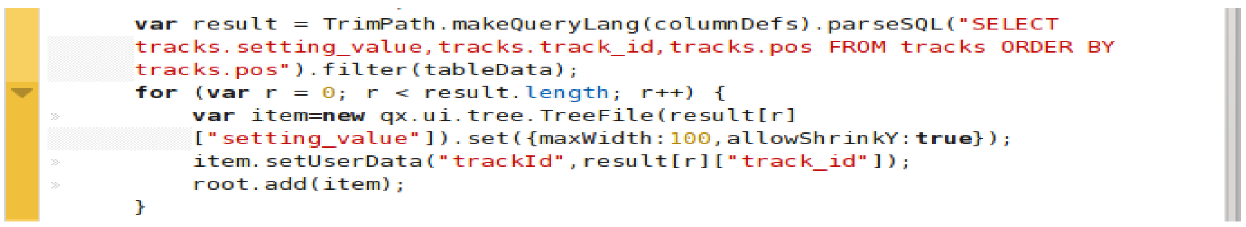

Menu Area
Info Area
11-th National Congress on Theoretical and Applied Mechanics
Home
Programme
Sessions
Authors
Search
About
National Committee of Theoretical and Applied Mechanics
Bulgarian Academy of Sciences
NCTAM2009

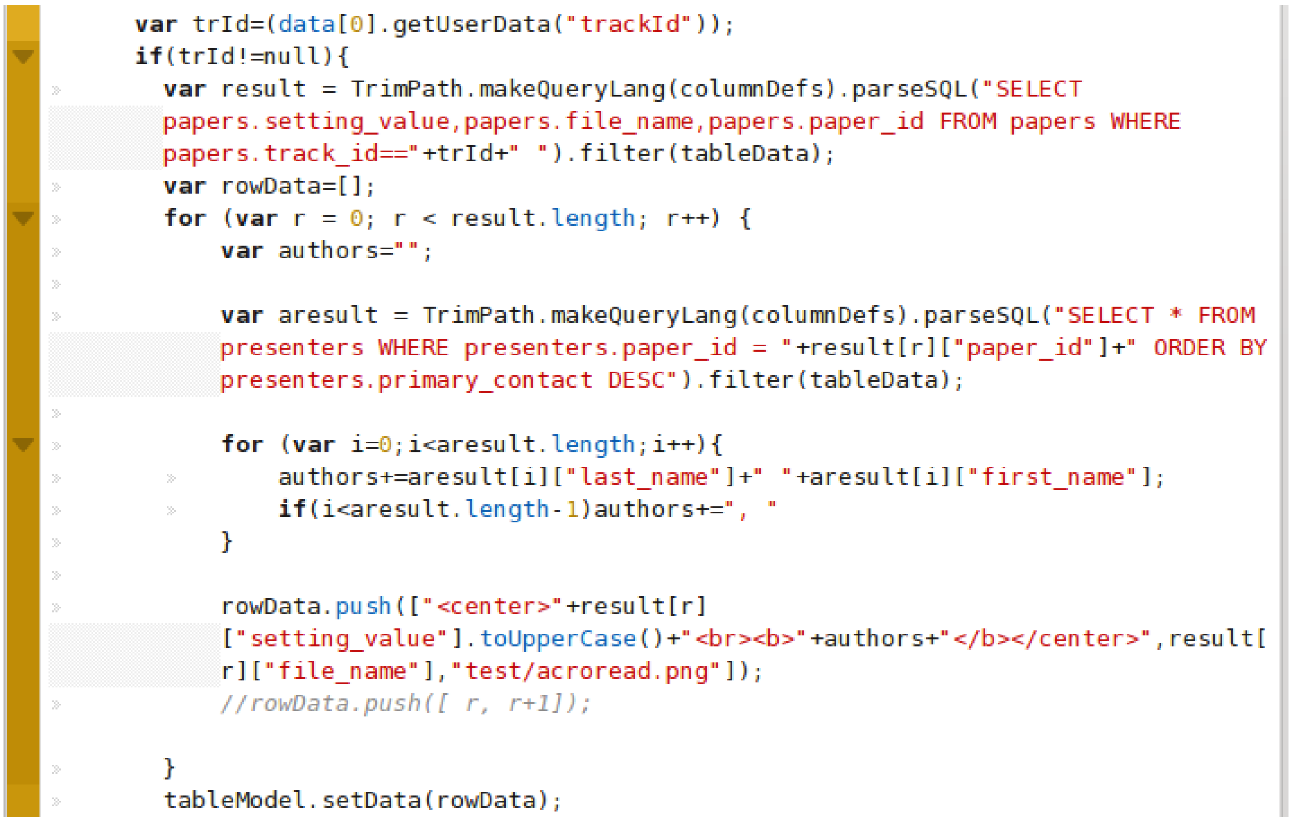
Fig.2 Main Window



```
        var result = TrimPath.makeQueryLang(columnDefs).parseSQL("SELECT
        tracks.setting_value,tracks.track_id,tracks.pos FROM tracks ORDER BY
        tracks.pos").filter(tableData);
        for (var r = 0; r < result.length; r++) {
            var item=new qx.ui.tree.TreeFile(result[r]
            ["setting_value"]).set({maxWidth:100,allowShrinkY:true});
            item.setUserData("trackId",result[r]["track_id"]);
            root.add(item);
        }
```

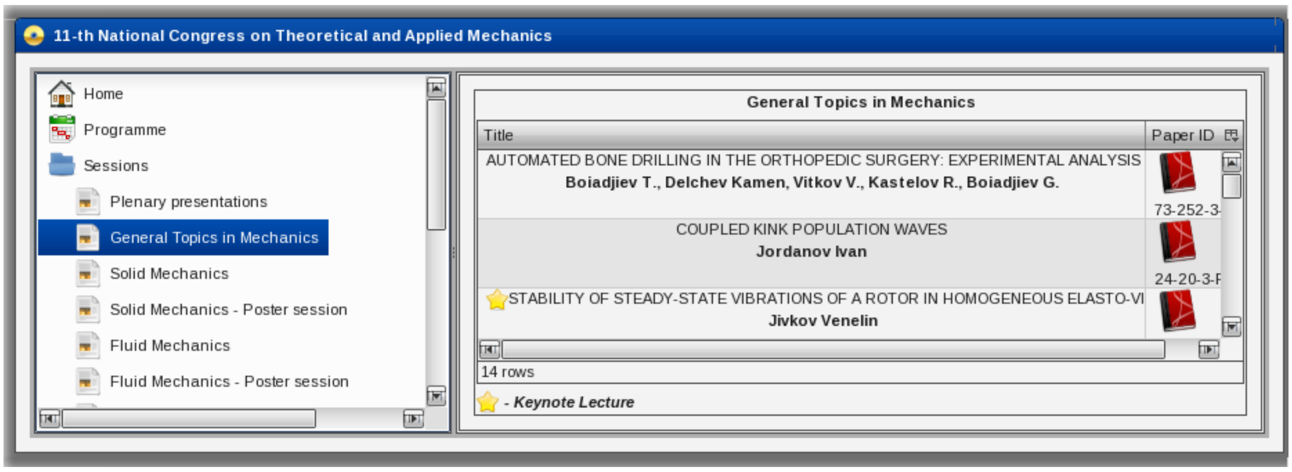
All menu items have assigned an identifier which is used in the events processing block to identify the correct command.

- **Home** and **Programme** menu blocks – display a static HTML page in Iframe inside the information area
- **Sessions** menu block – extracts the information for articles, authors and associated PDF file for the selected Session from the json like data base. This block uses the Table Model class from the qooxdoo framework with installed event listener on every row of the table. The code looks like the following segment :



```
    var trId=(data[0].getUserData("trackId"));
    if(trId!=null){
      var result = TrimPath.makeQueryLang(columnDefs).parseSQL("SELECT
      papers.setting_value,papers.file_name,papers.paper_id FROM papers WHERE
      papers.track_id=="+trId+" ").filter(tableData);
      var rowData=[];
      for (var r = 0; r < result.length; r++) {
          var authors="";

          var aresult = TrimPath.makeQueryLang(columnDefs).parseSQL("SELECT * FROM
          presenters WHERE presenters.paper_id = "+result[r]["paper_id"]+" ORDER BY
          presenters.primary_contact DESC").filter(tableData);

          for (var i=0;i<aresult.length;i++){
              authors+=aresult[i]["last_name"]+" "+aresult[i]["first_name"];
              if(i<aresult.length-1)authors+=", "
          }

          rowData.push(["<center>"+result[r]
          ["setting_value"].toUpperCase()+"<br><b>"+authors+"</b></center>",result[
          r]["file_name"],"test/acroread.png"]);
          //rowData.push([ r, r+1]);

      }
      tableModel.setData(rowData);
```

The view generated by this block is illustrated at the figure below:



11-th National Congress on Theoretical and Applied Mechanics

Home
Programme
Sessions
Plenary presentations
General Topics in Mechanics
Solid Mechanics
Solid Mechanics - Poster session
Fluid Mechanics
Fluid Mechanics - Poster session

General Topics in Mechanics

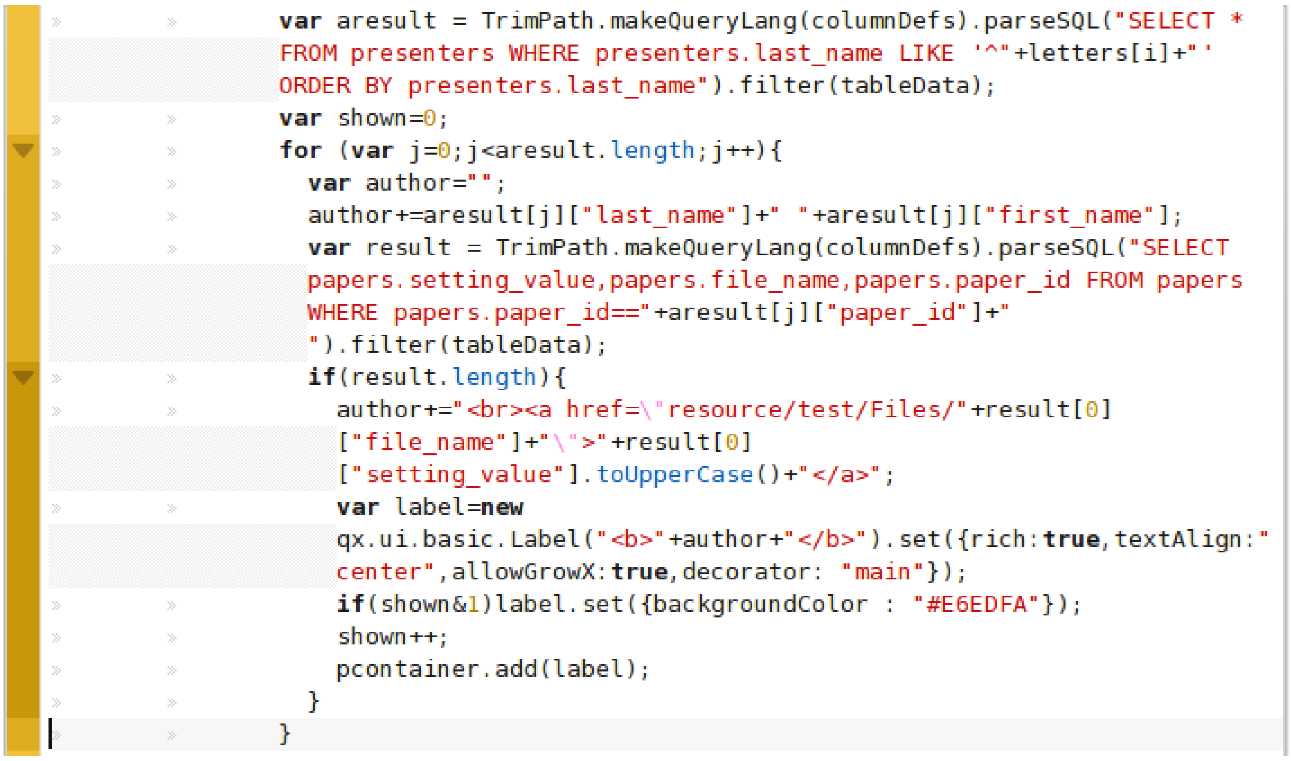
Title	Paper ID
AUTOMATED BONE DRILLING IN THE ORTHOPEDIC SURGERY: EXPERIMENTAL ANALYSIS **Boiadjiev T., Delchev Kamen, Vitkov V., Kastelov R., Boiadjiev G.**	73-252-3-
COUPLED KINK POPULATION WAVES **Jordanov Ivan**	24-20-3-F
⭐STABILITY OF STEADY-STATE VIBRATIONS OF A ROTOR IN HOMOGENEOUS ELASTO-VI **Jivkov Venelin**	

14 rows

⭐ - *Keynote Lecture*

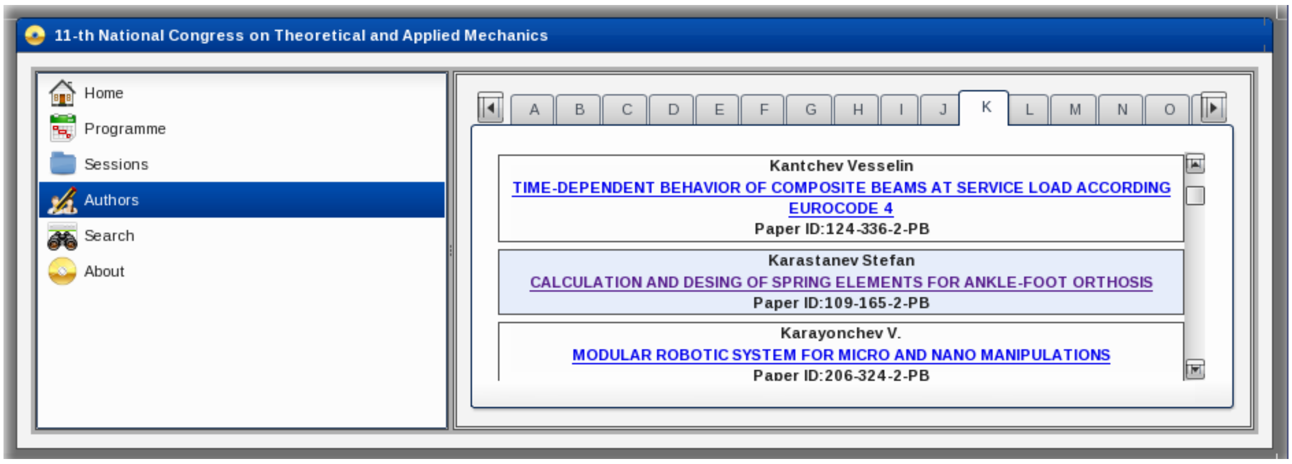
Fig. 3 Sessions menu

- **Authors** menu block – displays authors in alphabetical order. Extracts necessary data from json like database tables and shows the information in **TabView**[3] widget. The code is shown bellow:



```
var aresult = TrimPath.makeQueryLang(columnDefs).parseSQL("SELECT *
FROM presenters WHERE presenters.last_name LIKE '^"+letters[i]+"'
ORDER BY presenters.last_name").filter(tableData);
var shown=0;
for (var j=0;j<aresult.length;j++){
  var author="";
  author+=aresult[j]["last_name"]+" "+aresult[j]["first_name"];
  var result = TrimPath.makeQueryLang(columnDefs).parseSQL("SELECT
  papers.setting_value,papers.file_name,papers.paper_id FROM papers
  WHERE papers.paper_id=="+aresult[j]["paper_id"]+"
  ").filter(tableData);
  if(result.length){
    author+="<br><a href=\"resource/test/Files/"+result[0]
    ["file_name"]+"\">"+result[0]
    ["setting_value"].toUpperCase()+"</a>";
    var label=new
    qx.ui.basic.Label("<b>"+author+"</b>").set({rich:true,textAlign:"
    center",allowGrowX:true,decorator: "main"});
    if(shown&1)label.set({backgroundColor : "#E6EDFA"});
    shown++;
    pcontainer.add(label);
  }
}
```

The view generated by this block is illustrated at the figure below:


11-th National Congress on Theoretical and Applied Mechanics
Home
Programme
Sessions
Authors
Search
About
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O

Kantchev Vesselin
TIME-DEPENDENT BEHAVIOR OF COMPOSITE BEAMS AT SERVICE LOAD ACCORDING EUROCODE 4
Paper ID:124-336-2-PB

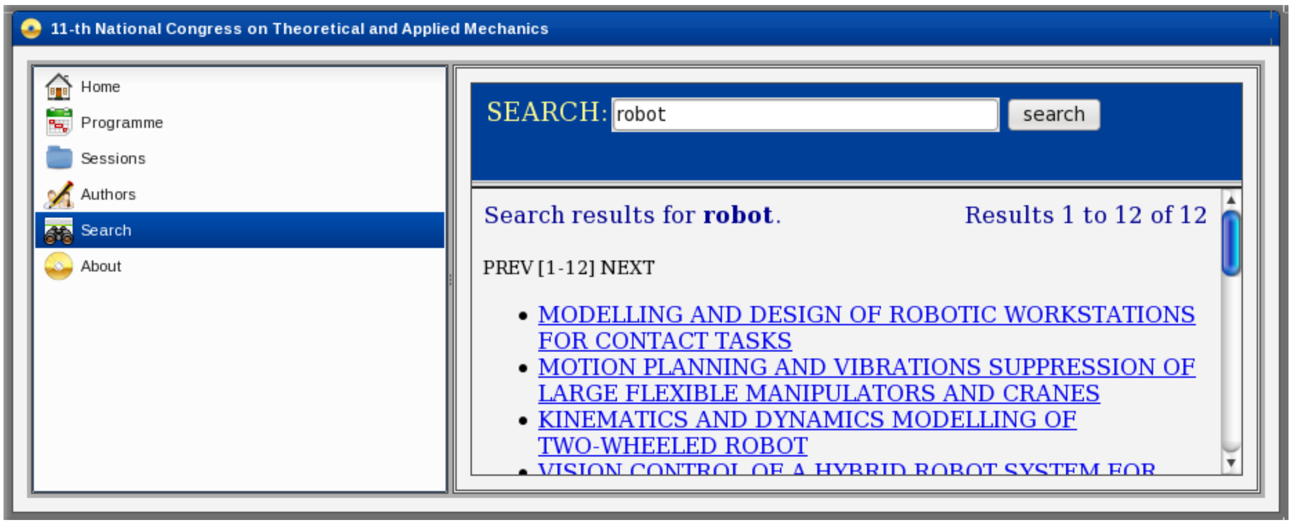
Karastanev Stefan
CALCULATION AND DESING OF SPRING ELEMENTS FOR ANKLE-FOOT ORTHOSIS
Paper ID:109-165-2-PB

Karayonchev V.
MODULAR ROBOTIC SYSTEM FOR MICRO AND NANO MANIPULATIONS
Paper ID:206-324-2-PB

Fig. 4 Authors menu

- **Search** menu block – deploys the full text search engine by integration into an Iframe inside the view area of the application. The embedded Iframe is prepared during the application initialization phase. The search scrip has been modified for better visual integration into the application.

Search block in action is shown at the figure bellow:


11-th National Congress on Theoretical and Applied Mechanics
Home
Programme
Sessions
Authors
Search
About
SEARCH:
robot
search

Search results for **robot**. Results 1 to 12 of 12

PREV [1-12] NEXT

- MODELLING AND DESIGN OF ROBOTIC WORKSTATIONS FOR CONTACT TASKS
- MOTION PLANNING AND VIBRATIONS SUPPRESSION OF LARGE FLEXIBLE MANIPULATORS AND CRANES
- KINEMATICS AND DYNAMICS MODELLING OF TWO-WHEELED ROBOT
- VISION CONTROL OF A HYBRID ROBOT SYSTEM FOR

- **About** menu block displays custom “About” information in popup tabbed window.

In addition to the main application of the CDROM builder system there is an optional PHP script, developed for the data extraction from particular online conference management system – Open Conference System. The script just executes properly crafted SQL queries to arrange the data from the online system in convenient form for Java Script SQL engine. The script downloads needed PDF presentation file as well.

All scripts described above have been developed and tested under Linux OS (Fedora Core 11). Tests in different environment have been done as well. The generated set of data has been tested for compatibility with different browsers (Mozilla Firefox, Google Chrome and Internet Explorer).

Conclusions

This study shows that it is possible to build an open source automated conference CDROM generator, based entirely on build-in facilities of the contemporary web browsers, without usage of any third party modules and programs. Tests which have been done show that the generated scripts provide modern user interface, stability and good performance under different web browsers and operating systems. Combination of different Javascript techniques ensures the flexibility of the system.

Acknowledgments

All data and logos used in this paper are copyrighted by Institute of Mechanics, BAS.

References

[1] Dojo Toolkit, http://www.dojotoolkit.org/docs

[2] Moo Tools, http://mootools.net/

[3] Qooxdoo, http://qooxdoo.org/

[4] TrimPath Query, http://code.google.com/p/trimpath/wiki/TrimQuery

[5] JSSIndex, http://jssindex.sourceforge.net/

Authors'information

Stefan Karastanev

Institute of Mechanics, Bulgarian Academy of Sciences, Acad. G. Bonchev Str., Bl.4,

Sofia 1113, Bulgaria

e-mail: stefan@imbm.bas.bg