M-2009

VIRTUAL LABORATORY: MULTI-STYLE CODE EDITOR

Andrey V.Lyamin, State University of IT, Mechanics and Optics St. Petersburg, Russia
Oleg E.Vashenkov, State University of IT, Mechanics and Optics, St.Petersburg, Russia

Summary

In general, virtual laboratory answer contain the description of some system with definition of input and output sets. This system is constructed on specialized virtual bench with conformance to entities and relations derived from subject domain. Knowledge about each right answer can be presented as input and output reference data set pair. To validate the answer it is necessary to obtain systems response with given reference input set and compare systems output with reference output set. System output and reference output coincidence will testify to answers correctness. Special virtual machine is used to obtain response of a system, which described in student answer, with given reference input set. The functions of informational-educational environment are virtual bench and virtual machine interaction, formation of tasks, reference data sets, logging of actions and results. This paper offers and describes original mechanism to build virtual laboratories in informational-educational environment AcademicNT. This environment has been developed at Saint-Petersburg State University of Information Technologies, Mechanics and Optics. The main achieved results are illustrated on a basis of laboratory "Multistyle code editor".

Introduction

Universities tendency to increase quality of training graduating students make conditional rejection of auditorium education towards individual education. However, requirement in mass education makes individual education ineffective. Solution of a problem is to transfer to electronic learning realized on specialized informational-educational environment. Such environment is built on informational technology to present and transfer learning materials. To achieve the best results, environment should provide complex approach in organization and quality evaluation of learning materials. Additional requirements are: virtual laboratory and training apparatus environment, adaptive learning dialogs, improvements of results monitoring and rating technology, wide communication possibilities of education process participants. Virtual laboratories in scope of informational-educational environment is necessary to form and evaluation of creative skills. Virtual laboratories distinguish from electronic test by presence of insoluble correct answers set. It is impossible to describe right answer for such tasks by finite set or formal expression (i.e. regular expression). Electronic tasks, built with virtual laboratories technology, are used in different disciplines, allowing to realize experiments and to control methods of implementation. The good example of virtual laboratory implementation is "Multistyle code editor", which means to evaluate algorithm building skills with multi-language programming possibility.

In general, virtual laboratory answer contain the description of some system with definition of input and output sets. This system is constructed on specialized virtual bench with conformance to entities and relations derived from subject domain. Knowledge about each right answer can be presented as input and output reference data set pair. To validate the answer it is necessary to obtain systems response with given reference input set and compare systems output with reference output set. System output and reference output coincidence will testify to answers correctness. Special virtual machine is used to obtain response of a system, which described in student answer, with given reference input set.

The functions of informational-educational environment are virtual bench and virtual machine interaction, formation of tasks, reference data sets, logging of actions and results. This paper offers and describes original mechanism to build virtual laboratories in informational-educational environment AcademicNT. This environment has been developed at Saint-Petersburg State University of Information Technologies, Mechanics and Optics. The main achieved results are illustrated on a basis of laboratory "Multistyle code editor".

1. Common interaction scheme

Virtual laboratory implementation provides interaction between different modules of informational-educational environment. After task request, AcademicNT transfers to client interactive element, often an applet (module on basis of Java Applets technology). Interaction between client and server is organized by HTTP protocol. Common tasks are represented as a web page optionally with applet. Answer submission is a responsibility of web page controls. Interaction between web page and applet is originated by web page, that requests user answer from applets data. Thus subject domain system, described by a student, is transferred to server as a text data. Web page transfers data to web server though HTTP protocol. Web server is a container for web applications. It transfers HTTP request to web application implemented as Java Servlet. Servlet prepares call to database stored procedure with parameters from request and that has been stored earlier. Evaluation laboratory server is realized as a separate application that listens custom TCP port for Remote Laboratory Control Protocol (RLCP) request. Stored procedure forms a special cursor to inform servlet on transfer of RLCP request to evaluation server, process response and a call another stored procedure for result preservation. RLCP message contains student answers, reference data sets and additional parameters. Evaluation server response is a XML RLCP message that contains result and time of execution for each reference data set pair. Servlet parses RLCP response and calls stored procedure for each result.
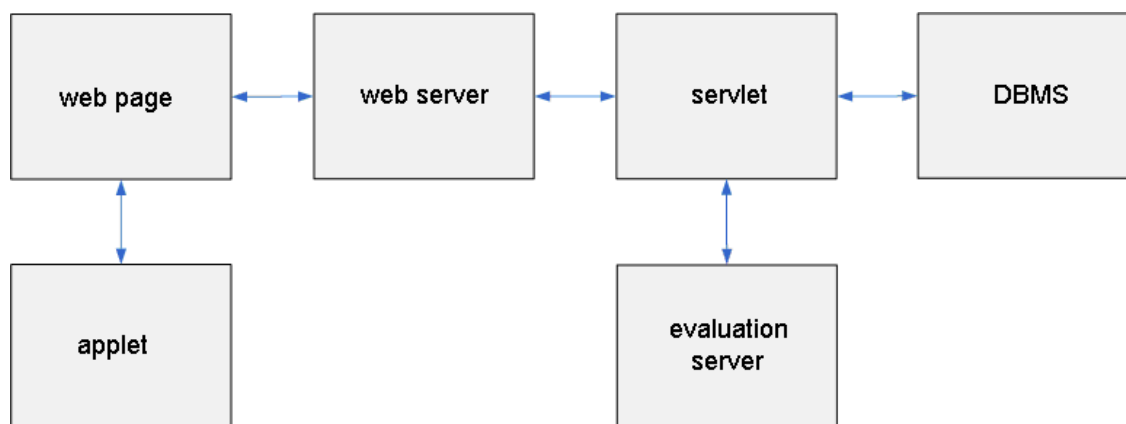The couples diagram is presented at Fig.1.

Fig.1 – Informational-educational environment modules interaction scheme

2. Algorithms of evaluation

Client request to web server casts to stored procedure call on DBMS. In general, stored procedures generate XML answer that translates into HTML of next web page. However, in case of laboratory answer evaluation, stored procedure returns a special command cursor instead of XML answer.

Evaluation can occupy long time and increase the load of system servers. Therefore, in some cases it is appropriate to suspend validation process. So when we have mixed test with simple question frames and virtual laboratory frames the simple frames evaluates immediately and laboratory frames evaluates later.

2.1 Immediate evaluation

Immediate evaluation can be used for testing scenarios when it is required to receive immediate result. However, in case of informatics and programming task, evaluation time depends on answer complexity. Sequence diagram of immediate evaluation is presented on Fig.2. Apparently from the figure, at immediate evaluation there are many interactions between system modules that lead to a considerable delay of response.

2.2 Deferred evaluation

Deferred evaluation allows answer preservation, evaluation execution scheduling and delivery of next question or test results. Immediate results will be calculated only for simple question frames, not a laboratory frames. The full statistics and results on laboratory will be accessible through means of monitoring of system. Sequence diagram of deferred evaluation is presented on Fig.3.

In AcademicNT execution time of deferred evaluation is distributed randomly between 5 and 60 minutes. Thus, it is provided simultaneously evaluation of deferred tasks while testing process that decrease a load of servers and response time.
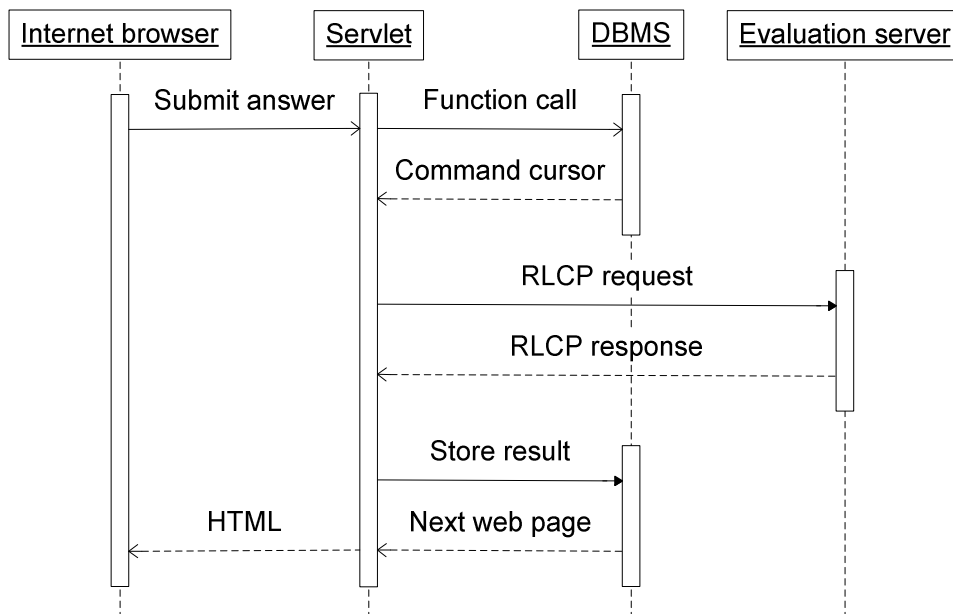
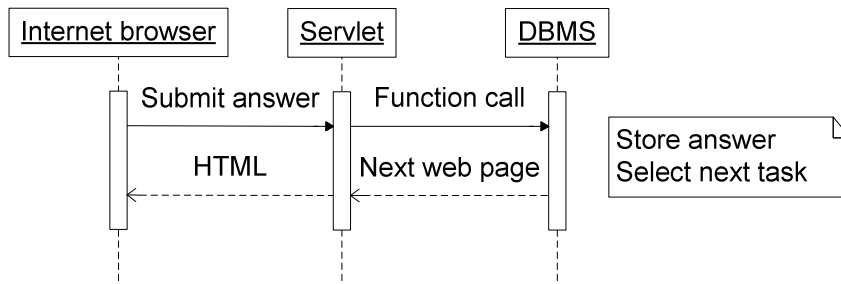Fig.2 – Sequence diagram of immediate evaluation process

Fig.3 – Sequence diagram of deffered evaluation process

The scheduler module is a part of AcademicNT web application. It is periodically polls DBMS for new tasks for which execution time has come. Tasks executes sequentially, therefore, actual time of execution can differ from the planned. Scheduler actions while deferred evaluation is presented at sequence diagram on Fig.4. Originator of activity is a Scheduler that chooses a new task with help of stored procedure. Stored procedure returns task description in XML document that contains all necessary data.

In error case, Scheduler terminates the task and holds it to later execution, after a time in web application configuration. After several errors, there is a sending of e-mail message to the system administrator.
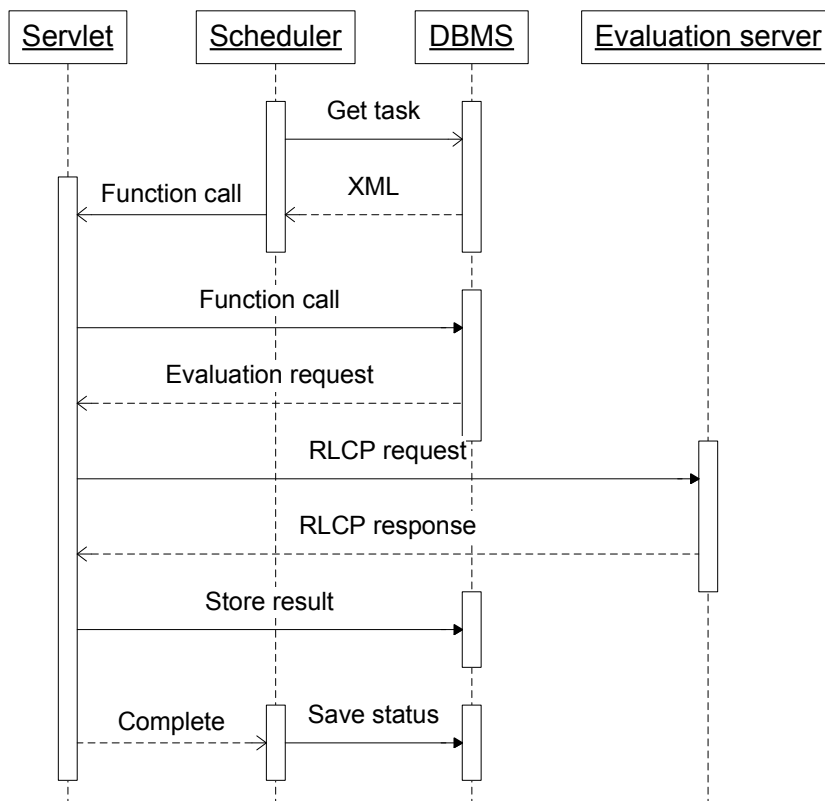

Fig.4 – Sequence diagram of execution of scheduling events

3. Remote Laboratory Control Protocol

The interaction between AcademicNT web application and evaluation servers is organized through RLCP protocol. This protocol was specially developed to provide a transition of data that is required for answer evaluation and for result presentation in XML format.
Sequence of evaluation is the following:
- Establishment of TCP connection;
- Sending a request from Servlet to Evaluation Server;
- Receiving response from Evaluation Server;
- Closing of TCP connection.

Request to evaluation server consists of header and body. Header is separated from body with a new line. The first line of header is a work mode (check or calculate). The next line defines fields conforms to format "name: value". The description of this fields presented in Tab.1. The Check mode means laboratory task evaluation and the Calculate means server calculations.

Table 1 – RLCP request fields

Name	Example	Description
url	url: rlcp://test:test@127.0.0.1:2005	Points an URL of evaluation server
content-length	content-length: 12345	Describes a size of RLCP-message (in bytes)

RLCP response has the following format:
- Response status code;
- Header fields;
- Empty line;
- Response body.

Response status is a response numeric code. Available response codes are described in Tab.2. Header fields are described in Tab.3.

Table 2 – Server response codes

Status code XXX	Description
100	The server continues to process request
200	Request is successfully processed and response contains results
300	Requested resource is temporary moved
400	Incorrect request
401	Incorrect reference sets
402	Authentication failed
403	Unsupported method (supported only Check or Calculate)
404	Absence of necessary header field
405	Absence of necessary header field value
500	Requested resource is temporary unavailable
501	Method is supported by protocol but not by this server

Table 3 – RLCP response fields

Name	Description
content-length	Size of body (in bytes). Only if when response code is 200

location	Points to real URL. Only if response code is 300

Each line in response or request should ends up with carriage return and new line symbols (with character codes 10 and 13 respectively).

RLCP message includes reference data sets identifiers, time of evaluation, laboratory output and boolean result for each data set. Evaluation server is a regular desktop application and can interact with external programs in any way.

XML-messages of RLCP have different structures for request and response. Request elements description is presented in Tab.4. DTD for RLCP request is a following:

```
<!ELEMENT Output (#PCDATA)>
<!ELEMENT Input (#PCDATA)>
<!ELEMENT Request (Conditions, Instructions)>
<!ELEMENT Conditions (ConditionForCalculating | ConditionForChecking+)>
<!ELEMENT ConditionForChecking (Input, Output)>
<!ATTLIST ConditionForChecking
        id CDATA #REQUIRED
        Time CDATA #REQUIRED
>
<!ELEMENT ConditionForCalculating (Input)>
<!ATTLIST ConditionForCalculating
        Time CDATA #REQUIRED
>
<!ELEMENT Instructions (#PCDATA)>
```

Table 4 – RLCP request elements

Element	Description
Request	Root element of message
Request/Conditions	Encapsulates reference data sets and evaluation parameters
Request/Conditions/ ConditionForChecking/@id	Reference data pair identifier
Request/Conditions/ ConditionForChecking/@Time	Time limit for evaluation (seconds)
Request/Conditions/ ConditionForChecking/Input	Reference input data
Request/Conditions/ ConditionForChecking/Output	Reference output data
Request/Conditions/ ConditionForCalculating/@Time	Time limit for calculation (seconds)
Request/Conditions/ ConditionForCalculating/Input	Reference input data set for calculations
Request/Instructions	User answer

Response XML elements are described in Tab.5. DTD for RLCP response is following:

```
<!ELEMENT Response (CheckingResult+ | CalculationResult)>
<!ELEMENT CheckingResult (#PCDATA)>
<!ATTLIST CheckingResult
        id CDATA #REQUIRED
        Time CDATA #REQUIRED
```

Result (0 | 1) #REQUIRED
>
<!ELEMENT CalculationResult (#PCDATA)>

Table 5 – RLCP response elements

Element	Description
Response	Root element
Response/CheckingResult	Container for results of single reference data pair
Response/CheckingResult/@id	Reference data set identifier
Response/CheckingResult/@Time	Time of evaluation
Response/CheckingResult/@Result	Boolean result as a an indication of coincidence reference data set with laboratory output
Response/CalculationResult	Result of calculations

4. Multistyle code editor

Multistyle code editor is virtual laboratory developed for evaluation of algorithm building skills. It is provides the following programming styles: Basic, C and Pascal.

Virtual laboratory applet allows to define variables, to write a code and to choose a programming style with template constructions available. Syntactically correct programs can be executed in applet at once or in "step by step" mode. Applets interface is represented in Fig.5.

There is a list of variables (with their values) in top frame. The "Edit" button allows modifying variables list or variable initial values. In the center the code edit area is represented with syntax highlighting option for chosen programming style. There is a library of template construction in the right frame. Double click at library item inserts corresponding template into code editing area. There is message pane in the bottom frame. Message pane accumulates compile, debug and runtime error and information messages. The list "programming style" is used for switching between programming styles that leads to change of a template library and syntax highlighting rules. There is a "free style" that is not allows for template construction, debugging and running programs but has syntax highlighting for most common keywords.

There are buttons for debugging and running control located at the bottom right side. Variable values changes are reflected at variables list while step by step debugging. The "stop" button allows to stop infinite loops. Evaluation server is watching for execution time from separate thread. After exceeding time limit, the answer is set off as wrong. The variables used in program, should be defined before execution or debugging. Dialog frame for variables operations is presented at Fig.6.
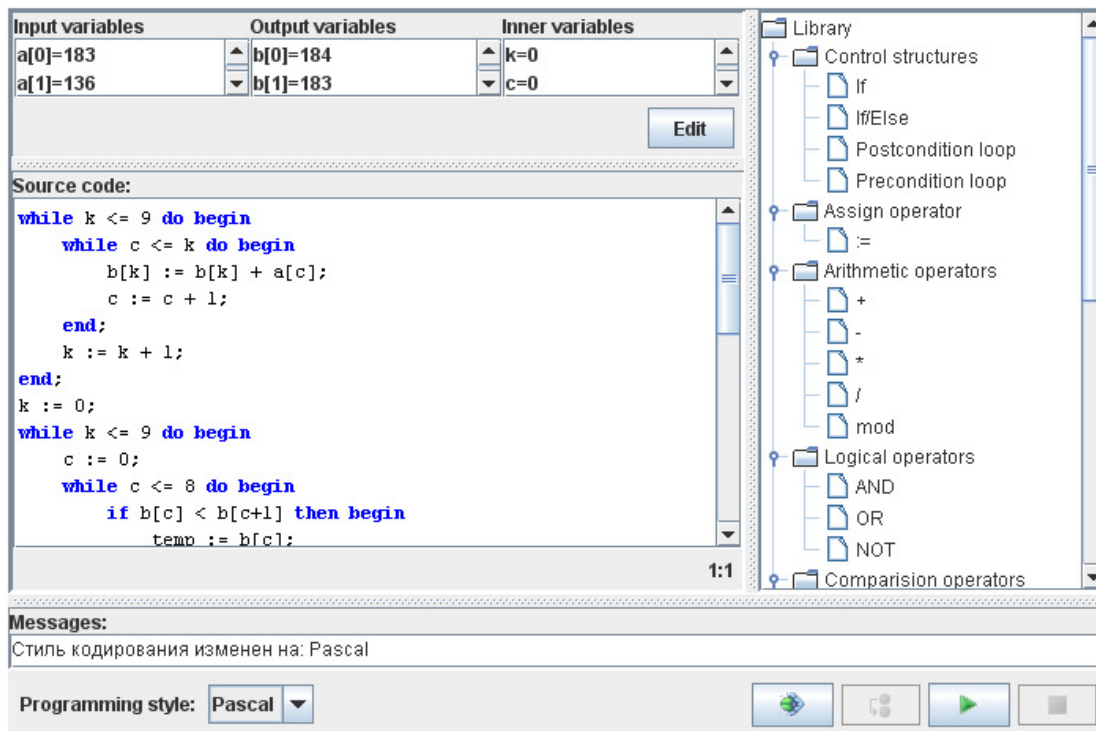
Fig.5 – Virtual laboratory applet interface

Variables can be set as input, output or internal. Input and output variables are used at automatic evaluation on a server. Values of input variables are established from a reference input set. Values of output variables are verified with a reference output set. Internal variables are used only for realisation of algorithms.

There are 5 types for variables: integer number, real number (with double precision), array of integers, array of reals and an array of characters (double byte unicode character set). Initial values of variables are set in a special dialog frame. In case of inconsistency of variable type and initial value, the error message is shown. Answer is considered as wrong if the server is disclosed such inconsistency while evaluation of a program.
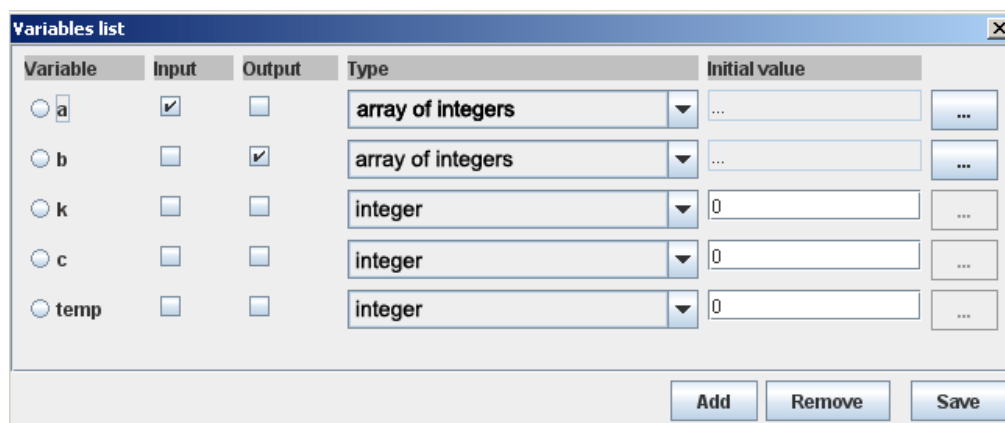

Fig.6 – Variables setup dialog

5. Data transferring issues

"Multistyle code editor" is one of the reference virtual laboratory implementation. It is used to algorithm building skills evaluation. From students it is required to prepare program code for concrete task and to define input and output variables. Applet encodes student answer as XML document during submission. DTD of this XML the following:

```
<!ELEMENT ProgramEnvironment (Variable+, Code)>
<!ELEMENT Variable (Name, Type, Value+)>
<!ATTLIST Variable
        input CDATA (true | false) "false"
        output CDATA (true | false) "false"
        inner CDATA (true | false) "true"
>
<!ELEMENT Type (#PCDATA)>
<!ELEMENT Value (#PCDATA)>
<!ATTLIST Value
        N CDATA #IMPLIED
>
<!ELEMENT Code (#PCDATA)>
<!ATTLIST Code
        style CDATA #REQUIRED
>
```

Answer is encoded in elements Name, Type, Value, Code as a comments. Therefore characters "-" take on special significance. XML parsing errors and evaluation stop is occurred without escaping. Escaping occurs in an order presented in the Tab.6.

Table 6 – special characters and corresponding entities

№	Character	Entity
1	&	&
2	<	<
3	>	>
4	-	&minus;
5	"	"
6	Line breaks (single characters and sequences of characters with codes 10 и 13)	

Conclusion

Informational-educational environment AcademicNT is intended for organization and realization of education process. One of the main advantages of the system is a virtual laboratory mechanism. It is used for formation and evaluation of skills. This paper describes modules interaction scheme, introduces evaluation algorithms and data exchange protocol. Main results are illustrated on a basis of "Multistyle code editor". This virtual laboratory is intended for algorithm building skills evaluation with programming possibility in the most widespread programming languages.

The results presented in this paper have passed approbation and have shown the efficiency during realization of following projects:
- Learning management system of SPbSU ITMO (http://de.ifmo.ru);
- Methodical Internet Center (http://ito-center.ifmo.ru);
- System for Internet-Olympiad and examination (http://de.ifmo.ru/exam).

In October, 2008 within the limits of 10th anniversary All-Russia forum "the Educational environment-2008" the All-Russia creative competition of scientific and technical decisions, educational products and services in the field of information of education was held. By results of competition information-educational AcademicNT environment has been recognised by the winner in a nomination "Corporate systems of remote training".