Automation of Administrative Work on the Arrangement of the Educational Process in the Smart E-Learning System Based on the Synchronization of the LMS MOODLE and 1C Information Base

Maxim A. Vikulin[a], Leonid L. Khoroshko[b] and Alexey L. Khoroshko[c]
Moscow Aviation Institute (National Research University), Moscow, Russia
vikulinma@mai.ru

Keywords: Synchronization 1C Moodle LMS e-learning.

Abstract: With all the advantages of smart e-learning systems, performance of administrative work on managing the student population and curricula creates a serious burden on the system operators. When performing all administrative tasks manually, there is a need for a large staff of operators to update data in a timely manner. On the other hand, it is possible to automate most of the tasks. Therefore, if an educational institution has an information system wherein the population data is stored and curricula are managed, the problem of student registration, enrollment in training courses, as well as updating data can be solved by synchronization of the administrative information system and the smart e-learning system. This paper discusses the development of a synchronizer and a 1C component for its implementation.

1 INTRODUCTION

The Educational Process Management Information System can be based on various platforms, but the 1C company software product and their 1C application configuration are in grate popularity 1C:University (Arguchincev, 2017). This configuration contains all the necessary information to automate the operation of the smart e-learning system, and the 1C platform has special tools for synchronization with other systems.

The smart e-learning system can also be implemented on various platforms, which have their own advantages and disadvantages. However, based on the statistics of the use of learning management systems (Abakumova, Bakaeva, Kolesina, 2016), the MOODLE platform is the most popular among others that provide the required functionality.

Due to the relevance of the use of smart e-learning (Capteraa, 2021) and the problems associated with the implementation of learning management systems in educational institutions (Pikalova, Lomakina, 2022), it was decided to automate a significant part of the support of smart e-learning systems in terms of maintaining the relevance of the student body and enrollment in training courses. So, it was decided to develop a software package that implements the synchronization of the 1C:University software product and the smart e-learning system based on the MOODLE platform. In this paper, the developed synchronization method and the 1C component are considered.

2 1C COMPONENT

The developed synchronization method is an Integration Toolkit software package, which is a package solution for synchronization of the 1C:University database and MOODLE-based smart e-learning systems. The main functionality is a system of linking disciplines from curricula to courses of smart e-learning systems. The software package also provides additional services for regulating the synchronization process and implementing the student's personal account (Khoroshko, Vikulin, Kvashnin, Kostykova, 2019).

[a] https://orcid.org/0000-0001-7885-1655
[b] https://orcid.org/0000-0002-7487-8997
[c] https://orcid.org/0000-0003-4434-0771

The Integration Toolkit software package consists of two interacting components: the 1C component and the MOODLE component.

The 1C component allows to link disciplines from the curricula to the corresponding courses of smart e-learning systems, and also collects all the necessary information for transmission to the MOODLE component.

The MOODLE component implements asynchronous gathering of information from the 1C component using a synchronizer, creates accounts for system users and enrolls students in training courses (Fig. 1).

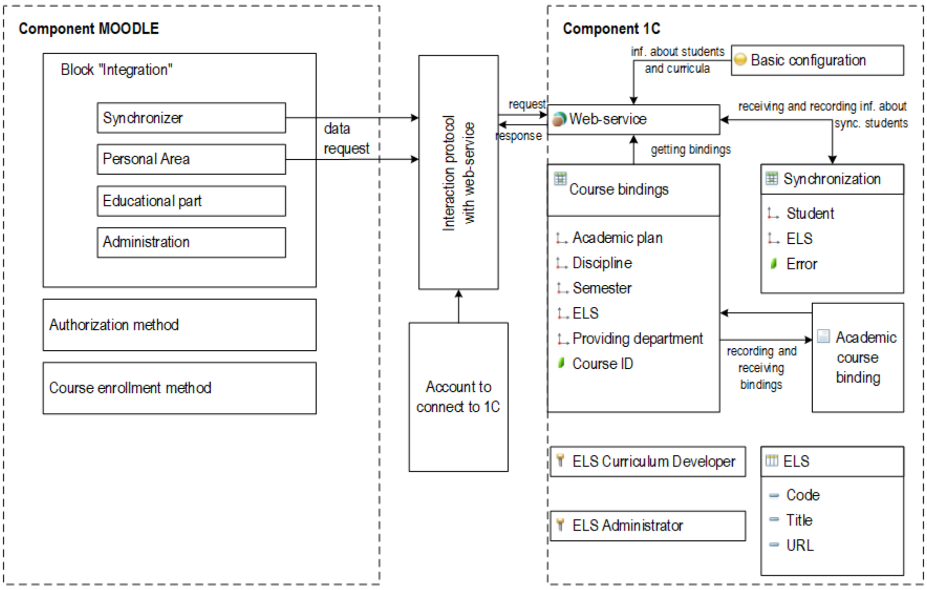

Figure 1: Software Package Architecture Diagram.

1C component consists of elements of the following configuration elements: Synchronizer web service, Synchronization data register, AcademicCourseBinding data register, AcademicCourseBinding document, ELS directory, ELSCurriculumDeveloper role, ELSAdministrator role.

All elements have the ITK_ prefix in the name for a clear visual distinction between configuration and search.

The web service is used to connect to 1C from smart e-learning systems (Aitov, 2015). When connected, the corresponding operations (functions) are called, to which parameters are transmitted.

The main function for getting student data is GetStudents (SystemName, Limit). The name of the smart e-learning system and the maximum number of records received by the request at a time are transmitted as arguments.

The first step of the function is to retrieve student records that have not yet been processed. The request selects the student's code, as well as links to the objects of an individual and a student. Students are received according to a cross-section of the latest statuses from the StudentStatus register, and students are selected only with the Continues Education status. In addition, student records are excluded from the sample, about which there is information in the Synchronization register, the period of which is greater than or equal to the current timestamp.

Next, the received students are sorted through in the cycle to obtain additional information:

- An academic group and a semester of study for a student are received from the StudentStatus register
- The last name, first name and patronymic of the student is selected by a slice of the last entries of the IndividualFullName register

- Information about study is selected from the StudyGroups directory: year of enrollment, field of study, profile, mode of study and site
- The information of the curriculum is obtained (the Curricula directory) on the correspondence with the group of the following parameters: the mode of study, year of enrollment, site, profile, field of study. The curriculum is also excluded from the selection if it is a template or the development status is not equal to the Approved status
- From the curriculum, the semester of study and the name of the smart e-learning system, non-empty bindings are obtained from a slice of the last entries of the AcademicCourseBinding register (the course identifier is nonzero). If bindings are found, they are collected in an array
- Upon successful receipt of all data (no errors occurred), they are recorded to the synchronization array, where the elements are structures with all the necessary information.

In case of errors when receiving data (the request does not return records), information about the error is recorded in the synchronization array and the transition to the next iteration of the cycle occurs. Errors, depending on the data, are of the following types: Status Error, Group Error, ES Error (executive software error), Individual Error.

After sorting through the student records, the generated synchronization array is encoded in JSON format. If a record in the array contains information about an error, then the structure for this student will contain only the "error" field, in the value of which the error text and the student's code will be specified. The SetErrorCronflag function will also be called. If there was no error, then a unified data structure will be formed for the student (Fig. 2).


```
{
   "st000000000":{
      "code":"0000000000",
      "stcode":"000000000",
      "secondname":"Ivanov",
      "firstname":"Ivan",
      "middlename":"Ivanovich",
      "group":"1V-601S-12",
      "semester":12,
      "plan":"160100 \"Aircraft building \" (2012/13)",
      "specialtycode":"160100",
      "specialty":"Specialist in Aircraft and Helicopter Engineering",
      "profile":"Aircraft building",
      "studyform":"part-time",
      "department":"101",
      "department_fullname":"Design and certification of aviation equipment",
      "faculty_code":"1",
      "faculty":"1",
      "faculty_fullname":"Aviation technology",
      "faculty_type":"Faculty",
      "area":"MAI",
      "courses":[
         1,
         2,
         3,
         4,
         5
      ]
   },
   "st000025613":{
      "error":"Group error (000020590)"
   }
}
```


Figure 2: Student Data Structure in JSON Format.

The SetErrorCronflag (Student, SystemName) function is an internal function of the web service. Sets the error information to the Synchronization register. Thus, the student will not be processed until either the error is corrected and the register entry is deleted manually, or the record period (one month) passes, after which the record is updated. The student's code and the name of the smart e-learning system are transmitted as arguments.

The SetCronflag (Students, SystemName) function records information about the successful synchronization of students in the Synchronization data register. As arguments, it takes the name of the smart e-learning system and a line of students in the following format: "StudentCode1#Period1, StudentCode2#Period2". The line is generated in the smart e-learning system, because the period is a timestamp in unix timestamp format.

The function splits the resulting line by separators, converts the period to the 1C time format, and then records to the register.

The Synchronization data register contains information about the synchronization of students. As measurements, a reference to the student object shall be specified (the Students directory of the main configuration) and the ELS object (the ELS directory). The Error resource containing a Boolean value is set to True if an error occurred while receiving student data and the student cannot be synchronized. The availability of this resource allows to quickly track students whose data errors need to be corrected, as well as restart synchronization for students with errors.

The register is independent, with a frequency within a second. The period records the date and time until which the student's record will be valid. In this way, the automation of the re-synchronization of students during the transition to a new semester is implemented. The synchronization period is set in the settings of the Integration unit.

The AcademicCourseBinding register contains information about bindings of academic courses. For binding, measurements consisting of discipline and ELS data as well as the identifier of the training course in the specified smart e-learning system are compared. The following measurements shall be specified: curriculum, discipline, semester, ELS and the teaching department. In connection with the architecture of the IASU configuration, a discipline can be a reference to an object from the following reference books: Disciplines, Cycles, Practices. The register resource is a CourseIdentifier containing a non-negative integer value.

The register is subordinate to the registrar with a frequency within a second. The registrar for the register is the ITK_AcademicCourseBinding document. The periodicity of the register allows to change bindings while preserving the change history (Khoroshko, 2019).

3 THE ACADEMIC COURSE BINDING DOCUMENT

The AcademicCourseBinding document is the registrar for the AcademicCourseBinding register and may be the basis for the implementation of the functionality for binding of academic courses. In order to store the history of document management, the surname and initials of the user who posted the document are stored in the Curriculum Developer account.

The tabular part of the document stores all the data necessary for registration in the register: curriculum, teaching department, discipline, semester, ELS and course identifier– as well as the Change and Several service fields for the correct operation of the document editing form (Alekseev, 2015).

Posting of the document triggers a movement mechanism by which the records of the tabular part with the Change flag set to the True value fall into the AcademicCourseBinding data register. Using the movement mechanism also allows to delete the corresponding register entries when canceling the document posting.

The document form is used for academic course binding. The form works in two modes: before and after the document posting.

Before posting the document (in particular, when creating it), a form is displayed designed for editing bindings. (Fig. 3). To work, you need to select an smart e-learning system, after which you can either display all bindings – both empty and filled - for the selected ELS (the Display Course Bindings button), or display only empty bindings (the Display Binding Free Disciplines button).

Data on disciplines and their bindings are displayed in a table with the following columns: sequence number, curriculum, teaching department, discipline, semester and course ID. All fields except the course ID are read-only. When clicking on any of the buttons for displaying bindings, the table will be cleared and filled in again, because all changes made by the user will be lost. In this situation, a warning message about data loss will be displayed. The user can either confirm the data loss and update the table, or select "Cancel" and then save the changes.

For operational work with course bindings in the form, there is a functionality improving usability (Radchenko, 2015). To select bindings, the user can use a filter containing fields divided into two blocks: Curriculum and Discipline. In the first block, it is possible to choose the curriculum itself, as well as narrow down the list by the graduating department, level of training and mode of study. The second block allows filtering by information related to the discipline. It is possible to choose a discipline, a teaching department, a semester, and also enter a numeric value of the course identifier.

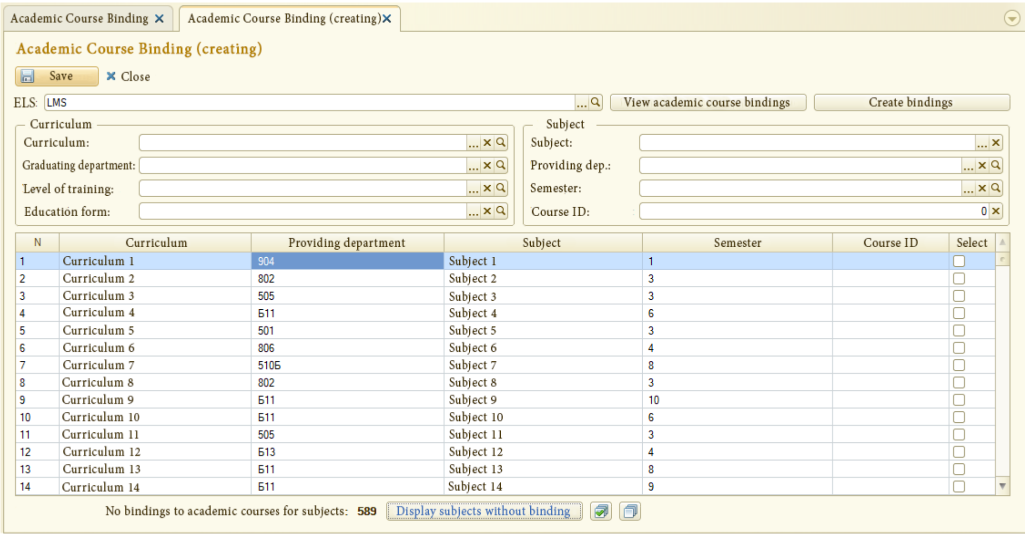

Figure 3: Academic Course Binding Form.

The displaying of the filtered list of disciplines allows to speed up the process of working with bindings not only due to the selection of information itself, but also due to the embedded functionality for implementing multiple bindings. The table has a column "Mult." (Binding Multiple Change), in which you can set the appropriate flag. When editing the course ID in any line that has this flag set, the change will be applied to all other lines that also have the "Mult." flag set. At the bottom of the form there are two flag setting buttons: Select All and Clear All, which allow to set or reset flags for all lines displayed in the table.

For informational purposes, the current number of empty bindings is displayed at the bottom of the form, that is, how many disciplines remain to be bound to academic courses (provided that all empty bindings have been created). Initially, the total number of empty bindings is displayed, but it is also calculated for each smart e-learning system separately – it is enough to select the desired ELS.

The form works only with the bindings themselves, which are stored in the AcademicCourseBinding data register. In this regard, an operation is required to create empty bindings, during which information from the curricula will be received and recorded in the register. At the same time, for an empty binding, the course identifier in the register is zero. Empty bindings can be created directly from the form, using the Create Bindings button. To create bindings, you need to choose a smart e-learning system. It is also recommended to select a curriculum so that the creation of bindings is performed only for the discipline of the selected curriculum. As a result, a corresponding document will be generated in which zero course identifiers will be set for all disciplines of the curricula. During creation, the existence of the binding is checked. Therefore, creating a new binding will not rerecord an existing one.

If the curriculum has not been selected, the creation of empty bindings will start in a cycle. At the same time, in order to save data, if a mistake is made in the curriculum, a separate document will be created for each curriculum. The operation to create empty bindings, depending on the number of curricula, may take a long time, up to the deletion of the session. In this case (as in the case of an error in the plan), the processed plans will be saved and the next run of creating empty bindings will be started for a smaller number of curricula.

To finish working with the form, there are two buttons: Save and Close. The Save button confirms the changes made and posts the document. At the same time, further editing becomes impossible for the curriculum developer. If an error was made in the bindings, the curriculum developer can either create a new document in which the necessary bindings will be rerecorded, or contact the ELS administrators to cancel the document posting.

If the document has already been posted (academic courses were bound), then all program control commands are hidden in the form and only a list of disciplines is displayed with all related data for which binding has been carried out. The data is displayed for reading only, editing is prohibited. This type of form is intended for viewing the binding change history (Malakhov, Galanova, 2019).

The ELS directory is used to store information about various smart e-learning systems. Therefore, the functionality of one-time interaction of single 1C configuration with several smart e-learning systems is supported.

The directory contains the standard attributes: Code and Name, as well as the Address attribute for storing the URI address of the smart e-learning system. In this case, the Name attribute is used to communicate with the MOODLE component: it must match the name of the smart e-learning system specified in the settings of the Integration unit.

The ELSCurriculumDeveloper and ELSAdministrator roles are used to grant access rights to configuration items. Roles have access to elements of the Integration Toolkit, as well as read access to those elements that are necessary for the functionality of the component.

The ELSCurriculumDeveloper role is given to employees responsible for binding of training courses. Their capabilities include the creation of new AcademicCourseBinding documents, with which they can create empty bindings and update them. The curriculum developer has Read Only access to all other elements. So, for example, a curriculum developer will not be able to create a new object of the smart e-learning system, but can link courses only to existing ones.

The ELSAdministrator role is necessary to administer the synchronization process. Has access rights to edit elements of the Integration Toolkit, as well as read access for related elements. The Administrator can create, edit and delete ELS objects, change the Synchronization data register in manual mode, as well as cancel the posting of training course binding documents. Also, this role is necessary for the account through which the smart e-learning system is connected to the web service.

4 CONCLUSIONS

A feature of the developed synchronization method is modularity. 1C:University component can be connected to the MOODLE smart e-learning system, which was taken as a basis, but each of the components can be changed. So, the implementation of the software package is also possible in universities using other 1C-based configurations, or using other learning management systems. Such changes require some adaptation to specific products and their versions, but the synchronization method will remain the same.

As an approbation and an example of the implementation of the developed method, we can take the Moscow Aviation Institute. The MOODLE platform is used as a smart e-learning system at the university, and the administrative information system is the Information and Analytical System of the University (IASU), which is an independently developed configuration for the 1C platform.

All continuing education students whose information is contained in the IASU have already been synchronized into the smart e-learning system and can receive credentials in the dean's offices.

As prospects for further improvement of the developed method of information system synchronization, it is planned to synchronize teachers into a smart e-learning system. At the moment, accounts for teachers need to be created manually, which leads to a decrease in productivity. In addition, each teacher must also be registered in the courses of the disciplines that he teaches. This process is also planned to be automated.

REFERENCES

Arguchincev, A.V., 2017. Platforma «1C: Predpriyatie» kak osnova postroeniya sovremennoj korporativnoj informacionnoj sistemy vuza. *Izvestiya Irkutskogo gosudarstvennogo universiteta. Seriya: Politologiya. Religiovedenie.* p. 121–131.

Abakumova, I.V. Bakaeva, I.A., Kolesina, K.Y. 2016. Technologies of initiating students into independent (self-guided) activity in supplementary distance learning. *IJCRSEE*, 2.

Capteraa, 2021. Most popular learning management system software. https:// www.capterra.com/learning-management-system-software/#infographic.

Pikalova, E.A., Lomakina, E.A., 2022. MOODLE AS A BASE FOR UNIVERSITY SYSTEM OF E-LEARNING AND FORMATION OF ELECTRONIC INFORMATION EDUCATIONAL ENVIRONMENT. *Problemy sovremennogo pedagogicheskogo obrazovaniya*, №74-2.

Khoroshko, L.L., Vikulin, M.A., Kvashnin, V.M., Kostykova, O.S., 2019. Communication with Students in Smart e-Learning System Using LMS Moodle. In: Uskov V., Howlett R., Jain L. (eds) Smart Education and e-Learning 2019. Smart Innovation, Systems and Tech-nologies, vol 144. Springer, Singapore.

Aitov, V.G., 2015. Integraciya informacionnoj sistemy vuza s sistemoj e-learning. *Prikladnaya informatika.*, №5 (59).

Khoroshko, L.L., 2019. Research of methods of synchronizing platform 1c with external information systems. *YUzhno-Sibirskij nauchnyj vestnik*, vol. 3, pp. 42–45.

Alekseev, A., 2015. 1C:Predpriyatie 8.3. Rukovodstvo razrabotchika. CH. 1.

Radchenko, M.G., 2015. 1C:Predpriyatie 8.3. Prakticheskoe posobie razrabotchika. Primery i tipovye priyomy: uchebnoe posobie.

Malakhov, Yu.A., Galanova, I.S., 2019. Ergonomic maintenance of designing intellectual user interfaces. *Ergodesign*, vol. 3, pp. 122-126.

Alekseev, A., 1C:Predpriyatie 8.3. Rukovodstvo razrabotchika. CH. 2.