

## ИНТЕГРИРАНЕ НА MOODLE С УНИВЕРСИТЕТСКИТЕ ИНФОРМАЦИОННИ СИСТЕМИ

*Силвия Гафтанджиева, Росица Донева*

*Пловдивски университет „Паисий Хилендарски“, гр. Пловдив, ул. „Цар Асен“ 24  
rosi@uni-plovdiv.bg*

**Резюме:** *Висшите образователните институции използват различни информационни системи за автоматизация на управлението, дейностите и процесите, вземането на решения и предоставяните образователни услуги. Широко разпространени са системите за студентска информация, средите за електронно обучение, библиотечните системи и др. Обикновено едни и същи данни се въвеждат ръчно в няколко системи, което предразполага към грешки. Наред с това, част от информацията трябва да се предостави и на други институции. Това налага необходимостта от изследване на подходите за интегриране на разнородни софтуерни системи, което е предмет на настоящата работа. Представено е решение и съответни експерименти за интеграция на CEO Moodle с информационните системи на висшите училища, основано на архитектура, ориентирана към услуги.*

**Ключови думи:** *софтуерна интеграция; интеграция на академични софтуерни системи; Moodle; архитектура, ориентирана към услуги; уеб услуги*

### 1. Въведение

Типично за съвременните висши образователни институции е автоматизиране на дейностите чрез голям брой взаимосвързани софтуерни системи за различни групи потребители: администратори, библиотекари, професори, асистенти, студенти, местни и глобални общности, изследователи, административен персонал и др. Използват се студентски информационни системи (СИС), библиотечни системи, дигитални архиви, системи за планиране и управление, за човешки ресурси, за оценяване и др. Развитие на интернет-базираните технологии и съответно на електронното обучение (е-обучение), доведоха до масово прилагане и на средите за електронно обучение (CEO) във висшите училища.

Поради това разнообразие, институциите са принудени да управляват сложна хетерогенна среда, в която влизат нови и наследени информационни системи, които изискват споделяне на данни помежду си, но обикновено са базирани на различен сървър, операционна система, платформа за съобщения, база данни и др.

В редица случаи, едни и същи данни се въвеждат многократно, което отнема време и предразполага към грешки. Така например, при провеждане на електронни курсове за обучение (е-курсове), оценяването на студентите се извършва в съответната CEO, а след приключване на е-курса, оценките на студентите трябва да бъдат въведени и в техните студентски досиета (в СИС). Друг пример – формираните в СИС данни при приема и регистрацията на новопостъпили студенти се налага да бъдат въвеждани отново, за да бъдат използвани в рамките на CEO или библиотечната система.

Преодоляването на подобни проблеми налага необходимостта от интегриране на отделните системи – решение, което позволява автоматизиране на процесите, премахва необходимостта от повторно ръчно въвеждане на данни и води до намаляване на възможностите за грешки.

Цел на представеното тук изследване е да се проучат съществуващите подходи за интегриране на разнородни софтуерни системи и да се предложи и експериментира решение, позволяващо интеграция на една от най-разпространените CEO – *Moodle*, с информационните системи на висшите училища.

## 2. Подходи и стандарти за интегриране на софтуерни системи

Проблемът, свързан с интеграция на софтуерни системи не е нов [11]. Той засяга както организации и компании, които имат нужда да интегрират своите вътрешни системи (вътрешно-бизнес интеграция), така и такива, за които е необходима външна интеграция с приложения на техни партньори, външни услуги и др. (бизнес-към-бизнес интеграция).

**Класическите интеграционни решения** попадат в категорията на софтуерните разработки, които функционират като допълнителен слой за трансация при взаимния обмен на информация между приложенията (т.нар. *middleware software*).

Друг разпространен подход е **интеграция на ниво потребителски интерфейс** чрез изграждане на портали, които обединяват функциите на различните приложения за крайните потребители.

Интеграцията може да бъде осъществена и на равнище данни чрез създаване на големи комбинирани масиви от данни. **Интеграцията на данни** обединява данни от различни източници и осигурява на потребителя единен изглед към тях. Сравнително ново направление в интеграцията от този тип са складовите данни, които осигуряват платформа за комбиниране и анализ на данните. Складовите данни събират и обработват данни от разнородни източници, дават възможност за анализиране на данните и взимане на управленски решения [2].

При интеграцията на данни широко се използват стандартизираните *EDI* (Electronic Data Interchange) транзакции, *XML* (Extensible Markup Language) документи, „плоски“ файлове (flat files) или електронни таблици.

Все по-налагащ се подход в разглежданото направление е т. нар. **архитектура, ориентирана към услуги** (АОУ) – *SOA* (Service-Oriented Architecture). Основна идея, осигуряваща интеграцията при АОУ е – функционалните възможности на дадена софтуерна система, които се налага да бъдат използвани от други системи, да се предоставят като набор от услуги, всяка от които е предназначена за решаване на конкретен проблем.

В основата на този подход са уеб услугите – софтуерни решения, които използват стандартизирани протоколи, позволяващи на отдалечени системи да общуват и да обменят данни помежду си. Уеб услугите обикновено се създават на отдалечени компютри (доставчици) и се извикват от клиентски приложения (потребители) по мрежата. За комуникацията между уеб услугата и клиентското приложение се използва *XML*. Уеб услугите използват стандартизирани протоколи - като *HTTP* (Hypertext Transfer Protocol), *SMTP* (Simple Mail Transfer Protocol), *FTP* (File Transfer Protocol) и др., което ги прави транспортно независими. В основата на уеб услугите стоят стандарти базирани на *XML* – *WSDL* (Web Services Description Language), *SOAP* (Simple Object Access Protocol), *UDDI* (Universal Description, Discovery and Integration) [12] и др. Стандартът *WSDL* се използва за описание на уеб услугите. В описанието на уеб услугите се съдържа информация за имената на функциите, типовете на данните, входните и изходните параметри на функциите, адреса за достъп до услугата и протокола, който се използва за обмен на данните. Обменът на информация между системите се осъществява чрез протоколите *SOAP* (създаден от Microsoft и приет от W3C), *REST* (Representational State Transfer), *XML-RPC* (XML-Remote Procedure Call) и др.

С развитието на „облачните“ технологии все по-широко приложение намира и съответният подход за софтуерна интеграция – **„облачната“ интеграция**. Той представлява процес на конфигурация на множество приложни програми за споделяне на данни в „облака“. В последните години „облачната“ интеграция печели популярност най-вече сред организации, корпорации и правителствени агенции, които прилагат разпределеният модел на изграждане на софтуер - известен под името „софтуер като услуга“ (*SaaS*, Software as a Service), при който приложенията се хостват от продавач или доставчик на услуги и са достъпни за организациите по интернет.

### 3. Избор на решение

От направеното проучване на съществуващите подходи за интегриране на разнородни софтуерни системи се налага изводът, че за целта на изследването е подходящо да се приложи АОУ, тъй като в сравнение с други

подходи за интегриране на системи има редица предимства, по-важни от които са:

- по същество е независим от това – как е конфигурирана останалата част от софтуерните решения на организацията, така че не изключва паралелно използване на портали, „облачни“ решения и складове данни;
- улеснява бъдещи промени на системите, които предоставят услуги в архитектурата – тъй като потребителските приложения са конфигурирани за достъп до услугите, без познания за предоставящата ги система, то основната система може да бъде заменена без това да влияе на „нейните“ потребители;
- гарантира по-голяма сигурност и защита, както и контрол върху данните на организацията, което е от особена важност за висшите училища и др.

Друго доказателство за приложимостта на АОУ при интеграция на софтуерните системи в дадена образователна институция е обстоятелството, че глобалната организация за стандартизация на образователните технологии – IMS Global Consortium, отрежда специално внимание на АОУ като средство за подобряване на оперативната съвместимост на образователните софтуерни решения (както вътрешни, така и външни за институцията), за реализиране на икономии чрез прилагане на отворени стандартни услуги и за привеждане в по-пълно съответствие на информационните услуги с услугите, които институцията предоставя на своята екосистема [1].

#### **4. Възможности за интегриране на CEO Moodle с външни системи**

Като основа за интеграция, в рамките на изследването е избрана CEO Moodle (Modular Object-Oriented Dynamic Learning Environment) – една от най-използваните системи с отворен код от образователни институции, бизнеса и отделни преподаватели.

В съответствие с новите тенденции на развитие на е-обучението, в последните версии на Moodle са предвидени възможности за интеграция с различни други програми, които тук са представени накратко.

Moodle осигурява **вградена интеграция с хранилища** [8], като *Google Docs, YouTube, Flickr, Alfresco, Box.net, Dropbox*, което предоставя на участниците в учебния процес възможност за лесно добавяне на документи от потребителските им профили в съответните хранилища. Добавянето на файлове от интегрираните хранилища е възможно след разрешаване от администратора на CEO Moodle (*Администрация на сайта/Добавки/Хранилища/Управление на хранилища*). След като необходимите настройки са направени, при добавяне на файлове потребителите виждат допълнителни

раздели с имената на хранилищата и след влизане в профилите им и разрешаване на достъп до данните имат достъп до наличните документи.

Възможно е и създаване на допълнителни модули, които да осигурят интеграция с други хранилища [9].

В SEO *Moodle* има възможност за създаване на потребителски профили, записване на потребители в учебни е-курсове и създаване на нови курсове чрез **интеграция с външна база данни** (БД), които съдържат такива данни [5, 6].

Регистрирането на потребители чрез копиране на данни от външна БД може да бъде включено от администратора на SEO, от раздела *Администрация на сайта/Добавки/Удостоверяване/Използване на външна база данни*. За осъществяване на връзката се задават: име на БД, име и парола на потребител с право на четене от БД, таблица с имена на потребителите, колони с потребителски имена и пароли, кодиране на данните и др. Настройките на връзката позволяват да се определи дали да се актуализира съдържанието на локалната БД на SEO *Moodle* при направени промени във външната БД и обратно и дали на потребителите да бъде забранено да променят стойности на полета [5]. При използване на този метод за регистриране на потребители, се прави проверка дали въведените потребителското име и парола са коректни. Потребителските данни се копират в локалната БД едва при първо влизане в системата, като се създава и потребителския профил.

Ако във външна БД се съхраняват идентификационни номера на курсове и техните потребители, то тя може да бъде използвана за записване на потребители в създадени *Moodle* учебни е-курсове. Настройките за достъп до такива записи от външната БД изискват задаване на тип, име и местоположение на БД, име и парола на потребител с права за четене от нея, имена и полета на таблици с потребители и учебни курсове. Записването на потребителите в курсове става автоматично при първото им влизане в SEO *Moodle*, при което от външната БД се получава информация в кои курсове и в каква роля да бъдат записани. Настройката на връзката с БД позволява да се избере възможност за автоматично създаване на нов курс в SEO *Moodle* при срещане на учебен курс, в който трябва да бъде записан потребителят, който все още не съществува в средата.

В *Moodle* е осигурена и **интеграция с други системи чрез уеб услуги**. Уеб услугите позволяват обмен на данни между системите чрез използване на протоколите *XML-PRC*, *REST* в *XML* и *JSON* (JavaScript Object Notation) формат, *SOAP*, *AMF* (Action Message Format). За целта, администраторът на SEO *Moodle*: активира възможността за използване на уеб услуги [10]; включва протоколите за обмен на данни с външни системи; генерира уникален код (нар. *token*) за свързване на потребителите с право да ползват услуги от външни системи и др.

Разработването на уеб услуга в *Moodle* [3,4] включва създаване на: файл за деклариране на предлаганите функционалности; файл с клас, дефиниращ уеб услугата (наследник на *Moodle* класа `external_api`), който включва по три функции за всяка функционалност (съответно за описание на нейната реализация, входните и изходните ѝ параметри и версията); и файл, съдържащ името на модула на уеб услугата.

## 5. Експерименти

От проведените експерименти за интеграция на СЕО *Moodle* с други системи на висшите училища тук са представени създадените уеб услуги за прехвърляне на оценки на студенти от *Moodle* към СИС и за автоматично създаване на е-курсове в *Moodle*, на базата на информация, получена от СИС.

### Експортиране на оценки от е-курс чрез уеб услуга

В СИС за всеки студент се пази информация за получените оценки по всички изучавани по време на обучението учебни дисциплини.

Учебните дейности в СЕО *Moodle* (тестове, блогове, уикита, форуми, задания и др.) позволяват поставяне на оценки на студентите в СЕО. При добавянето на дейност в рамките на учебния е-курс може да бъде определен максимален брой точки, който може да бъде получен при изпълнение на съответното задание или решаване на тест. Наименованията на всички задания и тестове, които ще допринесат за оформяне на крайната оценка на студента, заедно с получените резултати се включат в т.нар. отчет на оценките (*Grade Report*).

Крайната оценка се представя чрез събиране на точките получени от изпълнението на поставените задачи. За изчисляване на получената в края на провеждане на учебния е-курс оценка е необходимо получените точки да бъдат преобразувани в оценка от шестобалната система. За преобразуване на точките от отделните учебни дейности в крайни оценки се използват формули, подобни на формулите в системите за работа с електронни таблици, като всяко от дейностите трябва да има зададен идентификационен номер [7].

Върху крайните оценки оказват влияние всички учебни дейности, които са включени във формулите за изчисляването им дори и нулеви точки за учебни задания, които все още не са изпълнени или оценени (заданията и част от тестовите въпроси се оценяват ръчно от преподавателите). В резултат на това, е възможно при експортиране на оценките към външни системи да се прехвърлят некоректни оценки. Решението, което е използвано за преодоляване на този проблем е заключване на получените от студентите оценки в края на провеждане на курса и едва след това разрешаване на достъп до тях чрез уеб услугата.

В таблиците на БД на Moodle (табл. 1) се пази информация за записаните в курс студенти.

Таблица 1: Таблицы на БД с информация за записаните в курс студенти

Таблица	Съхранявана информация
<i>mdl_course</i>	Създадени в CEO курсове
<i>mdl_context</i>	Контекстите – система, курс, определена дейност
<i>mdl_role_assignments</i>	Зададени роли в различни контексти
<i>mdl_user</i>	Регистрирани в CEO потребители
<i>mdl_grade_grades</i>	Получени оценки за оценяваните дейностите
<i>mdl_grade_items</i>	Оценявани дейности (занятия, тестове, и т.н.)
<i>mdl_course_categories</i>	Създадени категории за курсовете

Създадената уеб услуга (фиг. 1) приема като входен параметър идентификационен номер на студентите, извлича от БД и връща като резултат: име и фамилия на студента, учебен предмет, получена крайна оценка. За да не се позволява изпращането на оценки от курсове, които все още не са приключили, е поставено ограничение заявката да връща само оценките на студентите, които са в интервала от 2 до 6 и са заключени.

```
SELECT c.id AS courseid, u.firstname AS first ,
       u.lastname AS last, u.idnumber AS number,
       c.fullname AS coursename, ROUND(gg.finalgrade,0) AS grade
FROM mdl_course AS c
JOIN mdl_context AS ctx ON c.id = ctx.instanceid
JOIN mdl_role_assignments AS ra ON ra.contextid = ctx.id
JOIN mdl_user AS u ON u.id = ra.userid
JOIN mdl_grade_grades AS gg ON gg.userid = u.id
JOIN mdl_grade_items AS gi ON gi.id = gg.itemid
JOIN mdl_course_categories AS cc ON cc.id = c.category
WHERE
  gi.courseid = c.id AND gi.itemtype = 'course' AND
  gg.finalgrade BETWEEN 2 AND 6 AND gg.locked>0
  AND u.idnumber = <идентификационен_номер>
```

Фигура 1. Заявка за експортиране на оценки от е-курс

### Автоматизирано създаване на учебни курсове

Уеб услугата за създаване на нови курсове от външна система в CEO Moodle автоматизира извършването на следните дейности:

- Създаване на е-курс за обучение;
- Добавяне на зададен брой теми към създадения е-курс като имената на отделните теми отговарят на теми от учебната програма на курса;
- Назначаване на преподавател на създадения е-курс.

Уеб услугата приема като входни параметри: пълното и краткото наименование на учебния курс, номера на категорията, към която ще бъде

добавен курса, броя и наименованията на учебните теми и името и фамилията на преподавателя. Необходимите входни параметри и техните типове, както и името на функцията за създаване на нов курс могат да се видят от WSDL файла с описание на уеб услугата (вж. фиг. 2).

```

<?xml version='1.0'?>
- <definitions targetNamespace='http://localhost:8080/moodle-2.2.5/moodle/webservice/soap/server.php?wstoken=ad737c87633edcf0c7bfe1a703c467d2'
name='webservives_virtual_class_000000' xmlns:wsdl='http://schemas.xmlsoap.org/wsdl/' xmlns:ssoap-enc='http://schemas.xmlsoap.org/soap/encoding/'
xmlns:xsd='http://www.w3.org/2001/XMLSchema' xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/' xmlns:tns='http://localhost:8080/moodle-
2.2.5/moodle/webservice/soap/server.php?wstoken=ad737c87633edcf0c7bfe1a703c467d2' xmlns='http://schemas.xmlsoap.org/wsdl/'>
- <types>
- <xsd:schema targetNamespace='http://localhost:8080/moodle-2.2.5/moodle/webservice/soap/server.php?wstoken=ad737c87633edcf0c7bfe1a703c467d2'>
- <xsd:complexType name='webservives_struct_class_000000'>
- <xsd:all>
- <xsd:element name='fullname' type='xsd:string' />
- <xsd:element name='shortname' type='xsd:string' />
- <xsd:element name='categoryid' type='xsd:int' />
- <xsd:element name='numsections' type='xsd:int' />
- <xsd:element name='sections' type='soap-enc:Array' />
- <xsd:element name='authorfirstname' type='xsd:string' />
- <xsd:element name='authorlastname' type='xsd:string' />
- </xsd:all>
- </xsd:complexType>
- <xsd:complexType name='webservives_struct_class_000001'>
- <xsd:all>
- <xsd:element name='fullname' type='xsd:string' />
- <xsd:element name='status' type='xsd:string' />
- </xsd:all>
- </xsd:complexType>
- </xsd:schema>
</types>
- <portType name='webservives_virtual_class_000000Port'>
- <operation name='local_coursesects'>
- <documentation>Used to create new course</documentation>
- <input message='tns:local_coursesectsIn' />
- <output message='tns:local_coursesectsOut' />
- </operation>
- </portType>
- <binding name='webservives_virtual_class_000000Binding' type='tns:webservives_virtual_class_000000Port'>
- <soap:binding transport='http://schemas.xmlsoap.org/soap/http' style='rpc' />
- <operation name='local_coursesects'>
- <soap:operation soapAction='http://localhost:8080/moodle-2.2.5/moodle/webservice/soap/server.php?
wstoken=ad737c87633edcf0c7bfe1a703c467d2#local_coursesects' />
- <input>
- <soap:body namespace='http://localhost:8080/moodle-2.2.5/moodle/webservice/soap/server.php?wstoken=ad737c87633edcf0c7bfe1a703c467d2'
encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' use='encoded' />
- </input>
- <output>
- <soap:body namespace='http://localhost:8080/moodle-2.2.5/moodle/webservice/soap/server.php?wstoken=ad737c87633edcf0c7bfe1a703c467d2'

```

Фигура 2: WSDL описание на услугата

При стартиране на уеб услугата, в СИС се прави проверка дали потребителят има право да я ползва. След проверка за валидност на входните параметри и правата на потребителя (трябва да има права за създаване на курсове) от входните параметри се извличат данните за пълното и краткото име на курса, категорията и броят на разделите (краткото име на курса и неговата категория са задължителни параметри при създаване на нови курсове като краткото име на курса трябва да бъде уникално в рамките на категорията, към която се добавя курсът). Следва извикване на вградената



функция на CEO Moodle за създаване на нови курсове, която създава курса като използва подадените входни параметри и връща като обект новия курс.

След създаване на курса се извлича неговия идентификационния номер и в таблицата от базата данни, в която се пази информация за темите на курса се добавят последователно всички теми от учебното съдържание, които са подадени като входни параметри на услугата.

В последната стъпка от изпълнението на уеб услугата се формира преподавател на курса. По зададените, като входни параметри, име и фамилия, от БД на Moodle се извлича идентификационният номер на преподавателя (след проверка дали потребителят, стартирал уеб услугата има право да задава роли) и му се вписва роля на редактиращ преподавател за създадения учебен е-курс, чрез активиране на вградената функция за записване на потребители.

След успешно създаване на учебния е-курс, в системата се връща информация за пълното име на курса и състояние „създаден“.

## 6. Заключение

Проведените експерименти доказват, че предложеното решение за софтуерна интеграция на базата на уеб услуги би подпомогнало образователните институции да предоставят по-гъвкава и навременна системна интеграция, която е по-малко податлива на грешки и по-рентабилна. В бъдеще изследванията ще бъдат насочени към предлагане на цялостно решение, основано на АОУ за предоставяне на интегрирано информационно обслужване във висшите училища.

## 7. Благодарности

Работата е частично финансирана от проект ИД13 ПО004 „Благополучие и психично здраве в кариерното развитие“ към Фонд „Научни изследвания“ при Пловдивски университет и финансираните от ОП „Развитие на човешките ресурси“ на ЕСФ проекти BG051PO001-4.3.04-0064 „Пловдивски електронен университет (ПеУ): национален еталон за провеждане на качествено е-обучение в системата на висшето образование“ и BG051PO001-3.1.08-0041 „Стандартизиране и интегриране на разнотипни информационни и управленски университетски системи (СИРИУС)“.

## Литература

1. Adoption of Service Oriented Architecture for Enterprise Systems in Education: Recommended Practices, Version 1.0 Final, September 2009.
2. J. Han and M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, 2000.

3. Moodle Doc, External Service Description, [http://docs.moodle.org/dev/External\\_services\\_description](http://docs.moodle.org/dev/External_services_description), посетен на 13.03.2013 год.
4. Moodle Docs, Adding a web service to a plugin, [http://docs.moodle.org/dev/Creating\\_a\\_web\\_service\\_and\\_a\\_web\\_service\\_function](http://docs.moodle.org/dev/Creating_a_web_service_and_a_web_service_function), посетен на 13.03.2013 год.
5. Moodle Docs, External database authentication, [http://docs.moodle.org/24/en/External\\_database\\_authentication](http://docs.moodle.org/24/en/External_database_authentication), посетен на 12.03.2013 год.
6. Moodle Docs, External database enrolment, [http://docs.moodle.org/23/en/External\\_database\\_enrolment](http://docs.moodle.org/23/en/External_database_enrolment), посетен на 12.03.2013 год.
7. Moodle Docs, Grade Calculations, [http://docs.moodle.org/20/en/Grade\\_calculations](http://docs.moodle.org/20/en/Grade_calculations), посетен на 12.03.2013 год.
8. Moodle Docs, Repositories, <http://docs.moodle.org/20/en/Repositories>, посетен на 27.03.2013 год.
9. Moodle Docs, Repository Plugins, [http://docs.moodle.org/dev/Repository\\_plugins](http://docs.moodle.org/dev/Repository_plugins), посетен на 12.03.2013 год.
10. Moodle Documentation, Web Services, [http://docs.moodle.org/dev/Web\\_services](http://docs.moodle.org/dev/Web_services), посетен на 27.03.2013 год.
11. Rikard Land, Ivica Crnkovic: Existing Approaches to Software Integration - and a Challenge for the Future, Fourth Conference on Software Engineering Research and Practice in Sweden, Linköping, Sweden, October, 2004.
12. W3C Working Group Note, Web Services Architecture, 11 February 2004, <http://www.w3.org/TR/ws-arch/#whatis>, посетен на 13.03.2013 год.

## INTEGRATING MOODLE WITH UNIVERSITY INFORMATION SYSTEMS

**Silvia Gaftandzhieva, Rositza Doneva**

*University of Plovdiv „Paisii Hilendarski“, Plovdiv, „Tzar Asen“ 24, rosi@uni-plovdiv.bg*

**Summary:** *High educational institutions use different information systems to automate management, operations and processes, decision making and educational services provided. Student information systems, e-Learning systems, Library systems, etc. are such examples. Same data usually is entered manually into multiple systems, which could lead to errors. In addition, part of information in a certain system has to be sent to other institutions. This imposes the need of examining the methods used for integration of heterogenic software systems that is the subject of the present work. An appropriate solution and corresponding experiments for integration of Moodle with information systems of the universities, based on service-oriented architecture, are proposed.*

**Keywords:** *Software Integration, Integration of University Information Systems, Moodle, Service-Oriented Architecture, Web-services*