Програмна реализация на средство за подготовка на упражнение в режим on-line

Ниязи Гарип, Ирина Желязкова

Program Implementation of a Tool for Exercise Preparing: *The work is the second of a sequence of papers concerning a teacher's tool for exercise preparing in on-line mode. Consequently in three separated paragraphs the tool's database relational model, programming modules, and algorithm for generating the exercise document are discussed.*

Key words: *Use Case Diagram, Activity Diagram, Sequence Diagram, Teacher, Tool.*

ВЪВЕДЕНИЕ

UML проектът на това средство е представен в предишна работа на авторите [3]. От **use case**, **sequence** и **activity** диаграмите в него става ясно, че приложението лесно може да се реализира с класическата трислойна клиент-сървър архитектура. Съдържанието в браузъра (първия слой) трябва да се променя в зависимост от данните, с които работи, и действията на преподавателя (П), т.е. да бъде динамично. Отличителна особеност на споменатия проект е, че информацията за дисциплините, упражненията и задачите има йерархичната структура, съответстваща на лекционния материал (ЛМ). Поради известните предимства и популярност на релационния модел на данните тази разнообразна по ресурси и голяма по обем информация е удобно да се съхранява в релационна база от данни (БД) - третия слой. За обработката и извеждането на информацията приложението има нужда и от среден слой, т.е. web-сървър плюс скриптове от страна на сървъра. Браузърът (клиентът) ще изпраща *HTTP* заявки към него, а той ще: записва/чете данни от таблиците на БД, извършва някаква обработка на данните и изпраща отговорите на заявките обратно към браузъра. Предвид казаното по-горе за програмната реализация на средството бе избрана известната тройка: Apache като web-сървър, PHP като език за програмиране и MySQL като система за управление на релационни БД. Предимствата на тази комбинация са добре известни от литературата и ИНТЕРНЕТ [5,6,9,10].

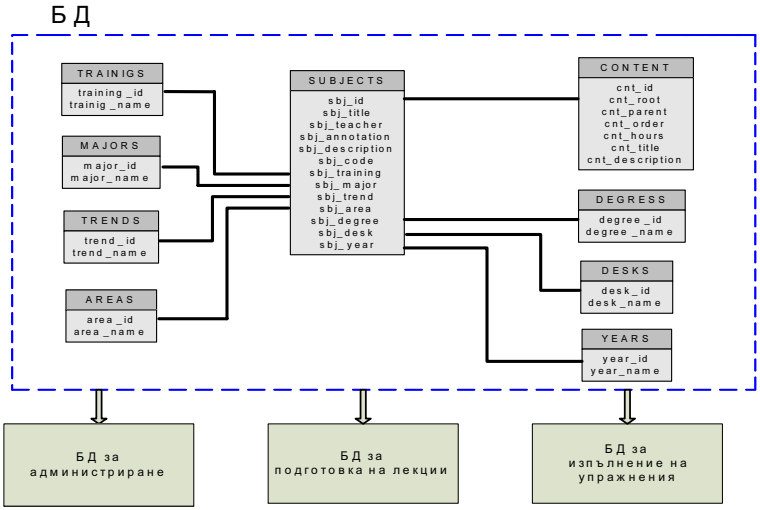
По-нататък тук е дадена програмната реализация с избраната технология на програмиране. Последователно в отделни параграфи се коментират релационният модел на БД, програмните модули и алгоритъмът за генериране на документа на упражнение. В заключението са дадени резултатите и намеренията на авторите.

РЕЛАЦИОНЕН МОДЕЛ НА БД

На фиг. 1. е показан синтезираният релационен модел на БД на средството. Освен това са дадени обобщено и връзките му с БД на три други модула съответно за: администриране, подготовка на лекции от водещия дисциплината и изпълнение на упражнения от обучавания. По-подробна информация относно цялостната стратегия на авторите за практическо обучение, представена чрез графов модел, може да се намери в [7,8].

Основни в БД на средството са таблиците **CONTENT** и **SUBJECTS**. В първата се съхраняват данни за всички задачи от базата с упражнения (БУ). В 7 на брой полета са данните за: упражнението, към което се отнася задачата (**Cnt_root** от целочислен тип); връзка с родителя на текущия възел-наследник (**Cnt_parent** от целочислен тип); подредбата на възлите-наследници от едно ниво на йерархия (**Cnt_order** от целочислен тип); хорариума на темата, т.е. брой часове (**Cnt_hours** от тип **tinyint** с дължина 2); наименованието на възела от дървото (**Cnt_title** с дължина до 255 символа); пълната информация за всеки един възел (**Cnt_description** теоретически с неограничена дължина).

Първото поле служи за връзка с таблицата **SUBJECTS**, в която се съхраняват данните за подготовката на упражнения. Предназначението на полетата (13 на брой) е следното: **Sbj_id -** идентификационният номер на упражнението, който автоматично се инкрементира, започвайки от 1; **Sbj_title** – темата на упражнението до 255 символа; **Sbj_teacher** – името на лектора на дисциплината от същия тип; **Sbj_description** – наименованието на дисциплината до 255 символа; **Sbj_code** – шифъра й в учебния план на специалността (от целочислен тип с дължина до 11 символа); **Sbj_annotation** – анотацията на дисциплината (теоретически с неограничена дължина); **Sbj_year** – годината на приемане на учебната програма (до 255 символа). Останалите шест полета (**Sbj_training, Sbj_major, Sbj_trend, Sbj_area, Sbj_degree, Sbj_desk**) служат за връзка със спомагателните таблици. Ще отбележим, че средството може да функционира и без поддръжката на данните в тези таблици.



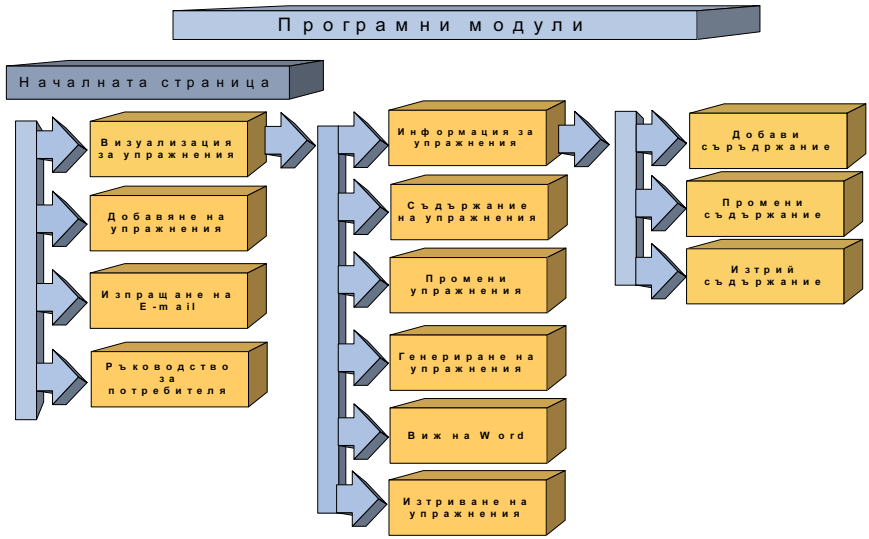
Фиг 1. Релационният модел на локалната БД заедно с други такива

В таблицата **DESKS** се съхранява информация за всички катедри като за идентификатор служи първото поле (**desk_id)** от целочислен тип с дължина до 11. Всеки следваща негова стойност представлява автоматично инкрементираната предишна стойност. Второто поле съдържа наименованието на катедрата (до 255 символа). Аналогичен е смисълът на първите полета за останалите таблици. Във второто поле на таблицата **TRAININGS** се съхранява видът на дисциплината (**задължителна**, **избираема**, **факултативна**), на **MAJORS** – наименованията на всички специалности, на **TRENDS** – наименованията на всички професионални направления, на **AREAS** – наименованията на всички професионални области, на **DEGREES** – степените (**бакалавър**, **магистър**, **докторант**), на **YEARS** – наименованията на всяка учебна година (от целочислен тип с дължина до 4).

ПРОГРАМНИ МОДУЛИ

На фиг. 2 е дадена йерархичната структура на програмните (***.php**) модули, от които е реализирано приложението. Първият модул отгоре надолу **Визуализация на упражнения** е без параметри и визуализира отляво под логото на Русенския Университет една таблица. В нея във вид на HTML списък се извеждат всички налични упражнения с информационното съобщение **Въведени упражнения**. Ако няма такива, се извежда съобщението **Няма упражнения.** Модулът **Добавяне на**

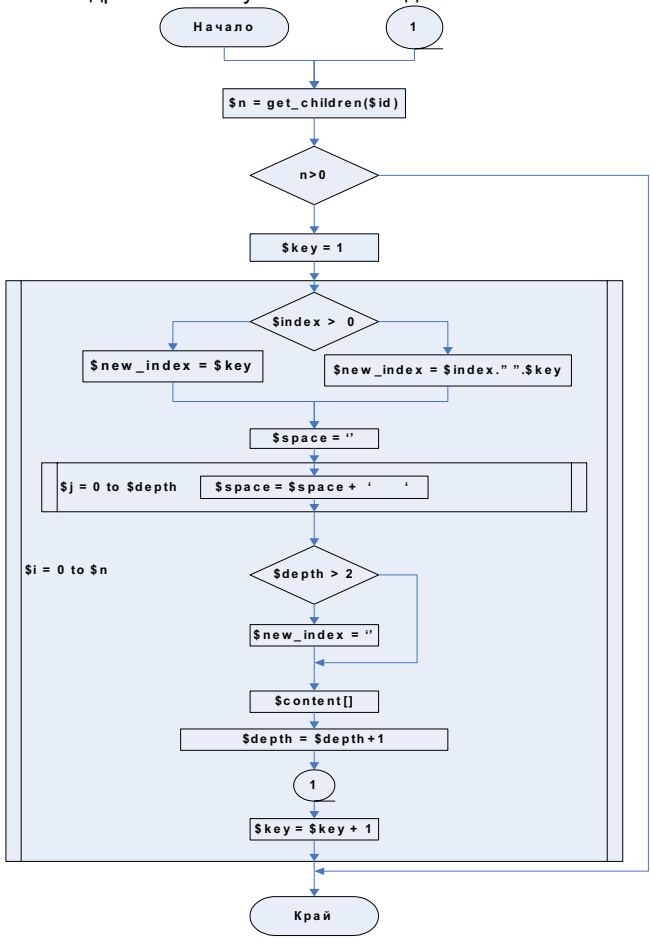
упражнения позволява да се добави ново упражнение като задължителните полета се маркират със символа '*'. От интерфейсна гледна точка представлява *HTML* формуляр с 10 полета от тип **combo-box** (**Заглавие**, **Преподавател, Анотация, Шифър, Тип обучение, Област** и **Квалификационна степен**) и 2 командни бутона съответно за добавяне и премахване на информацията от БД.



Фиг. 2. Йерархичната структура на програмните модули

Модулът **Информация за Упражнение** се извиква от предишния модул и по индентификационния номер на упражнението (***id***) само извежда наличната в БД информация за дисциплината. Следващият модул **Съдържание на упражнение** се извиква от главното меню и по идентификационен номер на упражнение формира масив от 6 елемента (***space***, ***key***, ***root***, ***content***, ***description*** и ***id***). Първите две полета зависят от дълбочината на задачата в йерархичната структура и представляват съответно генерираните празно място (***space***) и многопозиционен индекс (***key***). В дървовидна структура се визуализират всички задачи на избраното упражнение с възможност за добавяне на нова задача, редактиране на съществуваща, както и изтриването на отделните елементи на задача (формулировка, указания, решение, параметри, време за изпълнение и препратки към ЛМ. С модула **Модифицирай Упражнение** може да се коригират без триене и повторно въвеждане грешно въведени данни за упражнение. На външен вид е подобен на модула **Добавяне на упражнение** с тази разлика е, че полетата във формуляра вече са попълнени с въведените данни като се проверява дали в задължителните полета има данни. Модулът **Генерирай Упражнение** извършва най-сложната засега обработка на данните за избрано упражнение Резултатът представлява добре структуриран документ в зелено оцветен панел на синьо-сива основа. Модулът **Изтрий Упражнение** с входен параметър **$id** премахва от БД всички задачи и ресурси на указаното упражнение като предварително запитва дали П е сигурен в желанието си. С модула **Добави съдържание** се добавят нови задачи към упражнение чрез връзката родител-наследник. Представлява формуляр с 5 полета **(Упражнение, Наименование, Подредба, Време за изпълнение** и **Текст)** и 2 командни бутона. Подредбата посочва последователността, в която да се показват задачите на упражнение като при празно поле това става по датата на въвеждане.

За редактиране и форматиране на данните служи web-алтернативата на MS WORD готовият модул **fckeditor**. В отделни диалогови прозорци след въвеждане на определената информация той изпраща към сървъра генериран html код за запис в БД. Модулът **Промени съдържание** служи за отстраняване или модифициране на упражнение от БД по зададени идентификационен номер на дисциплина и на упражнение. Външно наподобява модула **Добави съдържание** с тази разлика, че полетата вече са попълнени. Модулът **Изтрий съдържание** изтрива задача от БД по зададен идентификационен номер като предварително запитва П дали е сигурен в желанието си. Последният модул **Изпращане на E-mail** служи за комуникация на студентите с П с входни параметри: e-mail на подател, получател, относно и текст на съобщение. Предварително проверява дали са въведени всички задължителни полета и дали e-mail адресът на получателя е валиден.



Фиг. 3. Алгоритъм за генериране на документа на упражнение

АЛГОРИТЪМ ЗА ГЕНЕРИРАНЕ НА ДОКУМЕНТА НА УПРАЖНЕНИЕТО

Блок-схемата на логическата част от алгоритъма е представена на фиг. 3. Използвани са две функции: **get_children()** и **get_subjects()**. Първата предварително филтрира елементите с родител упражнение с идентификатор (**$id**), брой задачи в

него (**$n**), брояч на задачите (**$key**) и брояч на упражненията (**$new_index**). Резултатът (**content[]**) представлява поддърво на първоначалното дърво. То се обхожда в дълбочина **($depth**) от втората функция чрез рекурсивното й извикване. При това се формират индексите (**$space**) на упражнението и задачите, включени в него. Резултатът от цялостния алгоритъм като външен вид, структура и съдържание на документа може да се види в [4].

ЗАКЛЮЧЕНИЕ

В тази работа е представена програмната реализация на средство за подготовка на упражнения в режим on-line. Обосновано е архитектурното и технологичното решение. Представени са релационният модел на БД, програмните модули, както и алгоритъмът за генериране на документа на упражнението. Потребителският интерфейс на средството може да се намери в [4].

Програмната реализация е в съответствие с представите и на други автори за структурирането на лекционния материал [1] и създаването на онтологии на обучаващите курсове [2]. Предстои реализацията на връзките по данни и управление с модулите за: администриране, подготовка на лекции и изпълнение на упражнения. Модулът за изпълнение на упражнения от обучавания ще се реализира с неголеми модификации в интерфейса и функционалността на модула за подготовка на упражнения.

ЛИТЕРАТУРА

[1] Вера Л., Использование ассоциативных связей для грануляции материалов курса, Proceedings of the Second International Conference “Modern E-learning”, Varna, Bulgaria, 2007, pp. 61-64.

[2] Еремин Е. О применении онтологий для представления программ учебных курсов, Proceedings of the Second International Conference “Modern E-learning”, Varna, Bulgaria, 2007, pp. 41-47.

[3] Гарип Н., Желязкова И. UML проект на средство за подготовка на упражнения в режим on-line, Трудове на научната конференция на Русенския университет, 2009.

[4] Гарип Н., Желязкова И. Потребителски интерфейс на средство за подготовка на упражнения в режим on-line, Трудове на научната конференция на Русенския университет, 2009.

[5] Кент А., Леа К. и др. PHP 4 for Beginners, 2002.

[6] Нарамор Е., Гернер Д. и др. Програмиране и Web дизайн с PHP5, Apache, MySQL, Том 2, София, “АлексСофт”, 2005.

[7] Желязкова И., Вълкова П. Стратегия за практическо обучение в задачно-ориентирани среди, Трудове на научната конференция на РУ, Том 44, Серия 6.1, 2005, pp. 107-112.

[8] Желязкова И. И. и др. Планиране и изпълнение на упражнения в задачно-ориентирани среди, Отчет по проект No.06-EEA-5, Русенски университет, Фонд “Научни изследвания”, 2006.

[9] http://bg2.php.net/manual/en/language.functions.php

[10] http://dev.mysql.com/doc/refman/5.0/en/functions.html

За контакти:

Доц. д-р Ирина Желязкова, Катедра “Компютърни системи и технологии”, Русенски университет “Ангел Кънчев”, тел.: 082-888 711, e-mail: Irina@ecs.ru.acad.bg

Докладът е рецензиран.