

UML проект на средство за подготовка на упражнения

Ниязи Гарип, Ирина Желязкова

UML Project of a Tool for Exercise Preparing: The work begins a sequence of papers concerning a teacher's tool for exercise preparing in on-line mode. Consequently in three separated paragraphs the tool's use case, sequence, and activity diagrams are presented and discussed.

Key words: UML, Use Case Diagram, Activity Diagram, Sequence Diagram, Teacher's Tool.

ВЪВЕДЕНИЕ

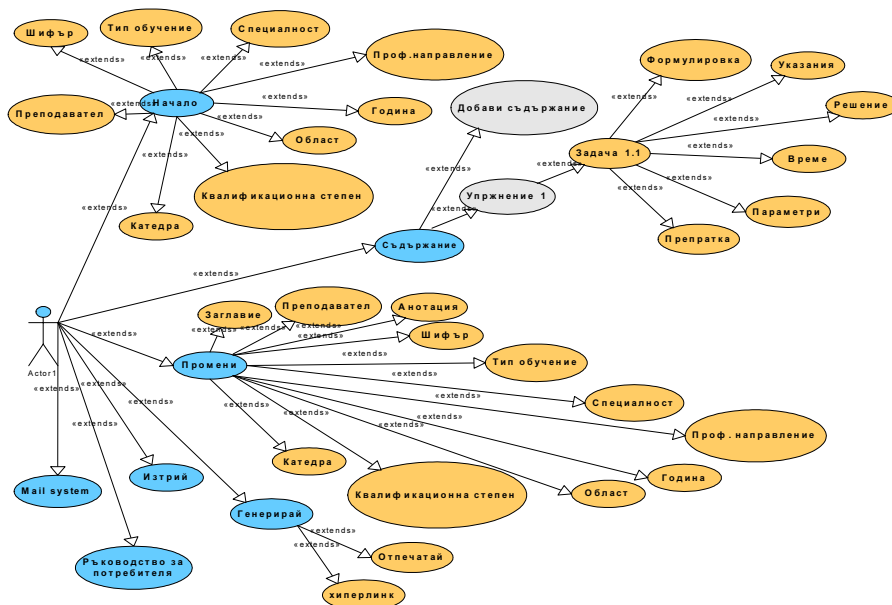
UML (Unified Modelling Language) вече е възприет като стандартен графичен език за визуализиране, специфициране, конструиране и документиране на софтуерни системи [9]. В актуалната версия на езика (UML 2.) са въведени 13 вида диаграми съответно на: класовете (**class**), компонентите (**component**), съставната структура (**composite structure**), разгръщането (**deployment**), обектите (**object**), пакетите (**package**), дейността (**activity**), машина на състоянията (**state machine**), случаите на употреба (**use case**), комуникацията (**communication**), преглед на взаимодействието (**interaction overview**), последователността (**sequence**) и времева (**timing**). В зависимост от спецификата на софтуерната система се препоръчва използването на различни техни комбинации, фокусиращи върху отделни нейни аспекти.

При проектирането на задачно-ориентирани среди (ЗОС) за електронно обучение, разработвани от колектив към Русенския университет, от тях се използват четири вида: use case, activity, sequence и class диаграми, които са езикови, технологични и платформено независими. Всяка ЗОС поддържа решаването на даден познавателен тип задачи и има четири категории потребители: системен администратор (СА), автор (А) на обучаващия курс (ОК), преподавател (П) и обучаван (О). Определена е строга йерархия в техните роли и права на достъп. СА може да играе ролята на А, П и О, А - ролята на П и О, а О - само своята роля. А е лицето, което в съответствие с целта на ОК, поддържа база от еднотипни задачи. П е отговорен за подготовката на упражнението, провеждането и оценката на изпълнението му от група обучавани. О, регистрирал се в ЗОС без потребителско име и парола, но с факултетен номер, се нарича трениран обучаван. Той може да наблюдава стъпка по стъпка демонстрацията на авторското решение на задачата, както и да запълва пропуските си в лекционния материал (ЛМ). О, регистрирал се с потребителско име и парола, се нарича изпитван обучаван. За да бъде оценен, той трябва да изпълни упражнението, виртуално управлявано от педагогическите умения на П и реално наблюдавано от същия. За подпомагане подготовката на упражнения бе предложена интеграция на две средства, работещи в off-line режим: текстов редактор [7] и зависим от типа на задачите редактор-генератор на програми [2]. Идеята бе реализирана и тествана за три вида задачи съответно за симулиране на динамични системи [12], построяване на алгоритми [13] и изчертаване на 3D фигури [8]. Алтернативен вариант на ЗОС може да се намери в [9], авторите на която са представили друга предметно-независима концепция на web-базирана система за практическо обучение. Известни са усилията и на други автори, насочени към създаването на подобни средства, но за нуждите на дадена дисциплина [5,11].

Тук се обсъжда UML проектът на средство, подпомагащо работата на П при подготовка на различни видове упражнения в режим on-line. Последователно в отделни параграфи се представят и коментират съответно диаграмите на: случаите на употреба (ДСУ), действията (ДД) и последователностите (ДП) за това средство. Заключение то съдържа основните резултати и бъдещи планове на авторския екип.

ДИАГРАМА НА СЛУЧАИТЕ НА УПОТРЕБА

Тази диаграма нагледно и на абстрактно ниво представя функциите на П (актьора), които средството трябва да подпомага. На ДСУ от фиг. 1 П играе само своята собствена роля. Предвид автономната засега работа на средството не е показан като актьор и ЗОС, за която средството ще изпълнява указаната услуга. Случаите на употреба (елипсите) по същество са действия, свързани с реализирането на on-line методика за подготовка на упражнение от еднотипни задачи. Освен за построяване на структурни схеми [1] и на блок-схеми на алгоритми, симулиране на динамични системи, изчертаване на 3D фигури, задачите могат да бъдат и за кодиране на програми.



Фиг. 1. ДСУ на средството от преподавателя

Плътните връзки между актьор и елипса означават, че П изпълнява основен потребителски случай. На фиг. 1 те са 5 на брой оцветени в синьо и наименовани (**Начало**, **Съдържание**, **Промени**, **Генерирай** и **Изтрий**). Пътят от актьор до крайна елипса се нарича сценарий (**scenario**). Сценариите са 30 на брой. Всяка стъпка в един сценарий е елемент на взаимодействие между актьора и средството. Използван е само един тип взаимодействие: **<<extend>>**, т.е. когато един случай на употреба използва част от функционалността на друг, за да разшири своята функционалност. Така, например, основният случай **Начало** отговаря на въвеждане на обща информация за ОК. Той използва допълнително случаите: **Преподавател**, **Шифър**, **Тип обучение**, **Специалност**, **Професионално направление**, **Област**, **Квалификационна степен** и **Категория**. Третият основен случай **Промени** се различава от първия по това, че отделните елементи на събраната информация се модифицират. Вторият основен случай **Съдържание** използва два допълнителни случая (**Добави упражнение** и **Упражнение с номер**), за

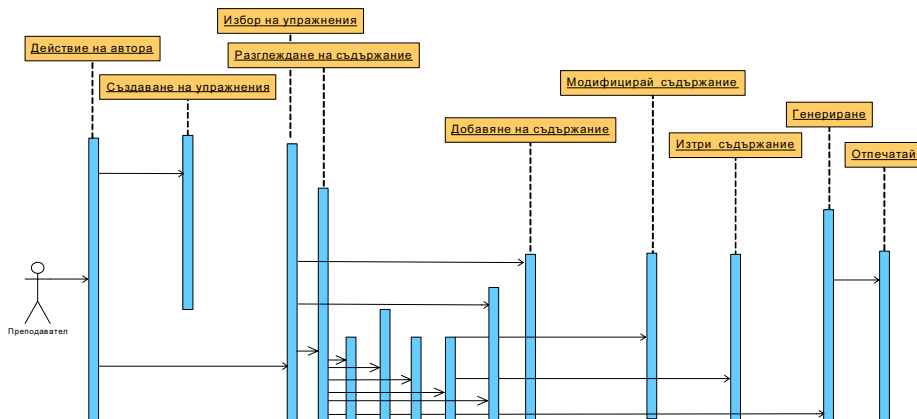
да добави информация за съдържанието на ново упражнение, и конкретна информация (**Формулировка, Указания, Време, Параметри, Решение, Препратка**) за всяка задача от упражнението. Четвъртият случай **Генерирай** се свежда до автоматично генериране на съдържанието на документа за избрано упражнение. Останалите също едностъпкови сценарии (**Word, Изтрий, E-mail и Виж**) са съответно за разпечатване на документ, изтриване на упражнението от базата с упражнения (БУ), изпращане на e-mail съобщения и разглеждане на ръководството за потребителя.

ДИАГРАМА НА ПОСЛЕДОВАТЕЛНОСТТА

ДП е използвана главно поради обектно-ориентирания и събитийно-управляван стил за програмна реализация на средството. На фиг. 2 действията на П, подпомогани от средството, са показани с хоризонтални правоъгълници. При това поради големия им брой действията, свързани със задачите, тук не са показани.

Вертикалните пунктирни линии са линиите на живот на действията. Потокът на времето е насочен отгоре надолу, което означава, че съобщенията на по-високо ниво се получават преди тези на по-ниско ниво. Всяко действие се изпълнява за определено време и в резултат от настъпването на определено събитие във вид на съобщение. Вертикален правоъгълник върху линия на живот означава активиране на съответното действие. С насочена отляво надясно стрелка се означава съобщението, което в дадения момент активира действието.

Ще отбележим, че серията от вертикални правоъгълници със стрелки, разположени в центъра на диаграмата, касаят отделните задачи. Липсват обратни връзки към вертикалните правоъгълници за задачите и упражненията, които означават повтарящи се действия. Предвижда се след окончателната програмна реализация на средството представената диаграма да се прецизира с наименованията на съответните методи над стрелките.



Фиг. 2. Диаграма на последователностите на средството

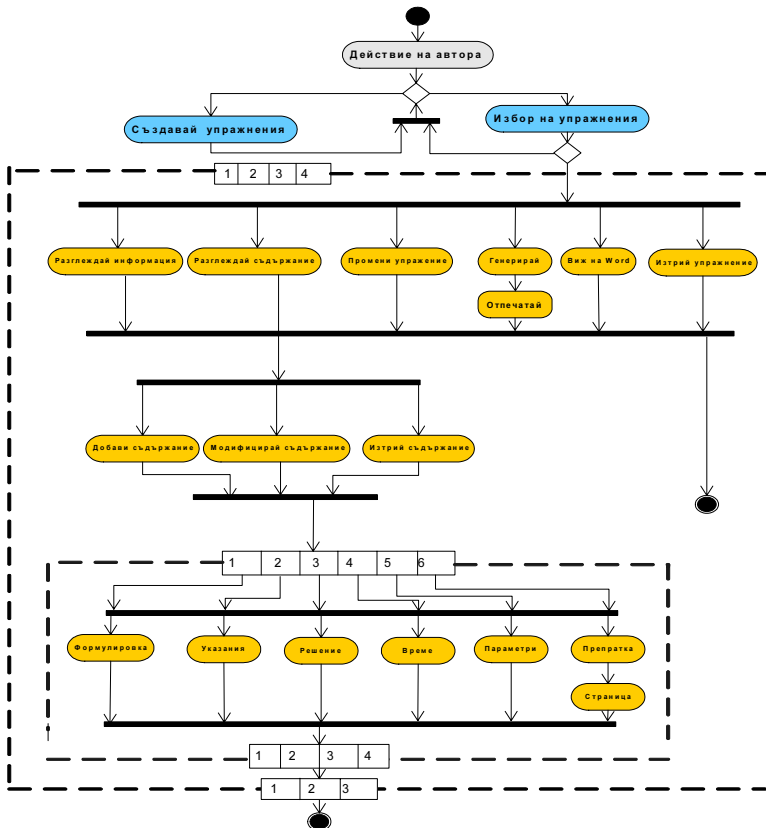
ACTIVITY ДИАГРАМА

Тази диаграма показва логическата последователност на действията, съставляващи дейността на П при работа със средството (фиг. 3). Диаграмата представлява комбинация от блок-схема на алгоритъм за вземане на решения и ординарна мрежа на Петри за отразяване на паралелните действия.

За означаване началото на информационния поток (**flow**) е използвана запълнена окръжност, а за края му - две запълнени окръжности, вложени в по-голяма окръжност. Вътрешните потоци от данни са означени със стрелки, а всяко действие със заоблен правоъгълник (**activity**). В него влиза един поток и излиза друг

поток. Ромбът (**decision**) се използва за класическите потоци от тип **if/then/else**. С вилица (**fork**) един поток се разклонява на няколко паралелни потока, когато са налице необходимите условия за това. С обърната вилица (**merge**) няколко потока от данни се съединяват в един общ поток.

Действията, свързани с подготовка на едно упражнение, са обединени в регион (правоъгълник с пунктирна линия). Действията, свързани с подготовката на една задача, са обединени в друг регион, вложен в първия. И в двата региона броят на пиновете в изходната колекция е по-малък от входната. Така в диаграмата е отразено търсенето на подходящи упражнения/задачи в БУ. Същите могат да се преизползват или модифицират за следващ сеанс с друга група от обучавани.



Фиг. 3. ДД на П със средството

ЗАКЛЮЧЕНИЕ

Чрез три популярни вида UML диаграми (use case, activity и sequence) е представен проектът на средство за подготовка на упражнения от П в режим on-line. То представлява алтернатива на off-line средство с подобна функционалност, вече тествано с различни типове задачи. Проектът отговаря на представата на други автори [6] за изпълнение на подпомагани от компютър задачи като многостъпков процес на преобразуване на символни ситуации. На първата стъпка О възприема и разбира формулировката на задачата, а всяка от останалите стъпки се представя с текущата символна ситуация, изпълняваното действие и резултатът от него.

Програмната реализация на средството подробно се коментира в [3], а потребителският му интерфейс - в друга работа [4]. Предвижда се в окончателната версия на представеното средство да се изчисляват параметрите на задачите (обем на знанията, степен на подсказка, степен на трудност) и да се регистрира времето за изпълнението им. Въз основа на тях ще се изчисляват и аналогичните параметри на упражнението, в т.ч. и скалата за оценка.

ЛИТЕРАТУРА

[1] Атанасова П., Бонева Й., Желязкова И. Програмна реализация на редактор-генератор на програми за структурни схеми, Трудове на научната конференция на Русенския университет, 2009.

[2] Борисов В. Интелигентен графичен редактор, Студентско научно творчество, Годишник, том XVIII, 2008, стр. 375-383.

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

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

[5] Димитрова С., Радойска П., Методика за генериране и решаване на задачи с електрически вериги в Web-базираны приложения, сп. Автоматика и Информатика, кн. 4/2005, стр. 60-63.

[6] Савельев А. Я., Новиков В. А., Лобанов Ю. И. Подготовка информации для автоматизированных обучающих систем, Москва, "Высшая школа", 1986.

[7] Стефанов С., Интелигентен текстов редактор, Студентско научно творчество, Годишник, том XVIII, 2008, стр. 219-223.

[8] Atanasova P. L., Stefanov S., Zheliazkova I. I. An Adaptive and intelligent Text Editor for Teacher's Support of Practical Exercises, Proceedings of International Scientific Conference Computer Science'2008, Sept. 18th-19th, Kavala, Greece, part II, pp.456-461.

[9] Dimitrova S., Lyubenov D. Three-Layer Architecture for Web-Based Practical Education, Second International Scientific Conference on Computer Science, Chalkidiki, Greece, 30th September-2nd October, 2005, part II, pp. 581-585.

[10] Fowler, M. UML Distilled: A Brief Guide to the Standard Object Modelling Language, 3rd edition, Addison Wesley, 2004.

[11] Ortega A., Álvarez R. Automating Exercises Validation in E-learning Environments, International Conference "Modern (e) Learning, Varna, 2007, pp.105-111.

[12] Valkova P. L., Levi A. S., Zheliazkova I. I. Preparing the Exercise Word Documents in a Task-Oriented Environment for Modeling and Simulation, *Proceedings of the First Int. Conference "Actual tasks and Multidisciplinary Sciences"*, Lozenetz, 6-8 June, 2007, Vol. 4, pp. 67-72.

[13] Zheliazkova I. I., Atanasova G., Practical Skills Acquisition in a Task-Oriented Environment for Algorithm Flowcharts Construction, Proceedings of International Scientific Conference Computer Science'2008, September 18th-19th, 2008, Kavala, Greece, part II, pp.462-467.

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.