

Погромська Г.С.

кандидат педагогічних наук, доцент;

Махровська Н.А.

кандидат фізико-математичних наук, доцент,

Миколаївський обласний інститут

післядипломної педагогічної освіти

УНІФІКОВАНИЙ ПІДХІД ДО ПРОЄКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

Однією з основних проблем при створенні великих і складних систем є проблема складності – технічна складність і складність управління. Моделювання є центральною ланкою всієї діяльності по створенню якісного програмного забезпечення. Моделі будуються для того, щоб зрозуміти та осмислити структуру і поведінку майбутньої системи, полегшити управління процесом її створення і зменшити можливий ризик, а також документувати прийняті проектні рішення. Сучасне бачення ролі моделей у розробці програмного забезпечення відображають рекомендації С. Амблера, які носять назву Agile Modeling [2].

У наш час для моделювання предметної галузі на ринку програмних продуктів представлено широкий спектр CASE-засобів. Найбільш популярними з них в нашій країні є Rational Rose, CA VPwin, Silverrun, Sybase PowerDesigner, Modelio. Оскільки, важливим, на наш погляд, є комплексність підходу та застосування єдиної уніфікованої нотації не тільки на етапі моделювання предметної галузі, але і на наступних етапах розробки програмної системи, тому найбільш доцільним засобом є Rational Rose [1; 6].

Характерним прикладом інформаційної системи, яка найповніше відобразить усі складові проектування класичної моделі, є інформаційне освітнє середовище. Така система реалізує на сучасному рівні функції не тільки освітнього процесу, але й управління ним та його якістю. Основна мета інформаційно-освітнього середовища університету полягає у забезпеченні можливості віддаленого інтерактивного доступу (в авторизованому режимі, орієнтованому на різні групи користувачів) до всіх освітніх ресурсів університету [3].

Продемонструємо методологію можливого підходу до опису інформаційно-освітнього середовища ЗВО із використанням мови UML.

Припустимо, що інформаційно-освітнє середовище, яке нами створюється (опис розглянемо у спрощеному вигляді), буде застосовуватися: викладачем (для створення нових курсів), студентом (для вибору курсу, що вивчається), менеджером (для загального управління навчальним процесом).

Моделювання інформаційно-освітнього середовища починається зі створення діаграми прецедентів (UseCase Diagram). Даний тип діаграми представлений акторами, елементами UseCase та відношеннями між ними [5].

Будь-який елемент UseCase представляє собою деяку частину функціональності, реалізованої в системі. Ці елементи можна ідентифікувати при розгляді кожної взаємодії актора з системою. У запропонованому нами прикладі: актор Student намагається отримати доступ до деякого курсу (Register); актор Tutor бажає запросити інформацію про особливості курсу, що викладається (Request); актор Manager повинен управляти різними організаційними питаннями (ManageEP (Manage Educational Process)). Таким чином, можна отримати наступну діаграму прецедентів.

Сценарії функцій предметної галузі можливо використовувати під час проектування сценаріїв роботи користувача з майбутньою системою, опис стану сутностей – для проектування користувацького інтерфейсу (довідника станів сутностей) та бази даних програмної системи. До того ж наявність сценаріїв функцій в подальшому дозволить уточнити функціональні вимоги до системи.

Більш детально функціональність елементу UseCase відображається в діаграмі послідовності (Sequence Diagram). Такі діаграми демонструють об'єкти та повідомлення між ними, які наочно представляють реалізацію поведінки. На рис. 1. надано приклад сценарію зарахування студента до деякого курсу, описаного із застосуванням діаграми послідовності дій UML (Sequence Diagram) для елемента UseCase Register. У пропонованому сценарії студент зараховується до курсу проектування програмних систем (PPS).

Опишемо сценарій, реалізований у діаграмі з рисунку 1. Студент Name повинен заповнити реєстраційну форму, після чого вона стає доступною менеджеру (на діаграмі зазначене представлено через взаємодію студента Name з реєстраційною формою regform). Далі форма посилає повідомлення менеджеру manager. Manager отримує повідомлення, що Name повинен бути доданий до певного навчального курсу (наприклад, курсу PPS). Для обслуговування даного запиту

повинен існувати об'єкт-курс PPS. Якщо він існує, то студента Name потрібно додати до цього курсу. Об'єкт-курс не має повноважень додавання студентів. Ця дія є прерогативою екземпляра класу CoursePreparation. У діаграмі послідовності він носить назву cp1. Якщо cp1 відкритий, то об'єкт-курс звертається до нього з запитом додати студента Name. Після чого студент отримує доступ до курсу, який він вказав у реєстраційній формі.

Об'єкти, представлені за допомогою діаграм послідовності, групуються в класи. Виходячи з розглянутої діаграми, можна ідентифікувати наступні об'єкти та класи: regform є об'єктом класу RegForm; manager – об'єктом класу ManagerEP; PPS – об'єктом класу Course; cp1 – об'єктом класу CoursePreparation.

Побудова діаграми класів є наступним етапом у процесі опису інформаційно-освітнього середовища на мові UML [5].

Після того, як додані основні класи, потрібно коректно вказати відношення між ними. Для того, щоб відобразити, яким чином об'єкти мають взаємодіяти один з одним, вивчаються діаграми послідовності та визначаються шляхи взаємодії між ними.

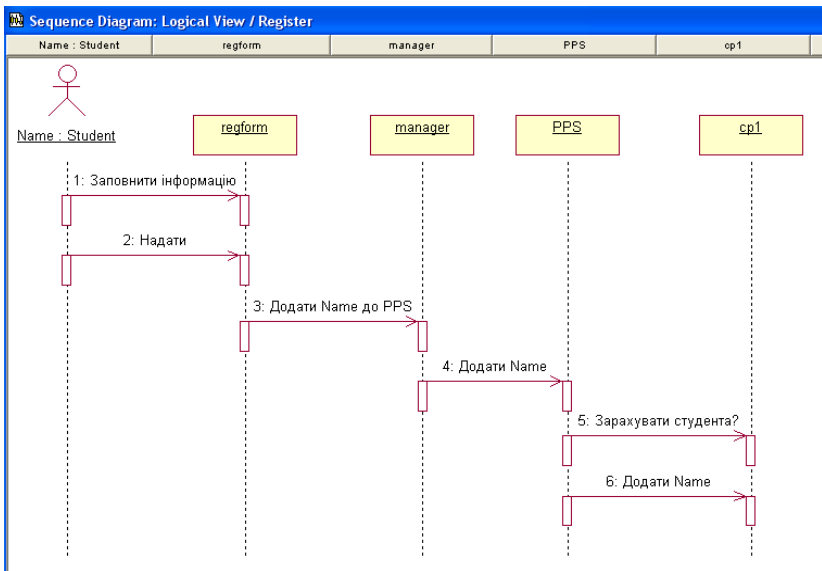


Рис. 1. Діаграма послідовності

Двома типами відношень є асоціація та агрегація. Асоціації визначають шляхи між об'єктами однакового рівня. Агрегація описує нерівноправні зв'язки. У пропонуваному прикладі існують асоціації (від RegForm до Manager та від Manager до Course) та відношення агрегації (між класом Course та класом CoursePreparation, так як ПідготовкаКурса є частиною класу Курс).

Будь-який клас включає деяку структуру даних та поведінку. Наприклад, клас CoursePreparation має один атрибут numberStudents та дві операції Add() та OpenPreparation(), які описують його поведінку. Таким чином, діаграма класів закономірно матиме наступний вигляд (рис. 2).

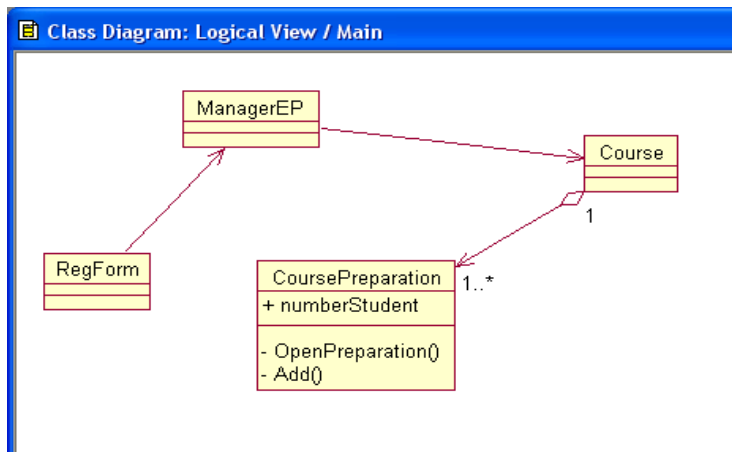


Рис. 2. Діаграма класів

Описана модель наведена для демонстрації процесу проектування і тому є неповною. Вона може містити ще багато структурних компонентів. Наприклад, якщо у процесі опису моделі інформаційно-освітнього середовища потрібно генерувати коди для класів, то слід скористатися компонентним поданням середовища Rational Rose (Component View). У розглянутій моделі буде один компонент CoursePr для опису файлу специфікації за класом CoursePreparation та ще один компонент CoursePrep для опису файлу реалізації.

Авторами розглянута лише умовно-спрощена модель опису інформаційно-освітнього середовища. Запропоновану модель можна

розширити, описавши процеси зі зв'язками між різними підрозділами середовища під час вирішення конкретних практичних завдань (горизонтальні зв'язки на діаграмі діяльності (Activity Diagram), що відбувається перед створенням діаграми варіантів використання UML (UseCase Diagram)) та змодельовавши документи.

Опис предметної галузі із застосуванням UML добре сприймається експертами предметної галузі. Таким чином, представлені елементи опису інформаційно-освітньої системи на базі мови UML дозволяють зробити висновок про достатню легкість і разом з тим гнучкість під час застосування подібних технологій для проектування сучасних інформаційно-освітніх систем.

Список використаних джерел:

1. Арлоу Дж. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование / Дж Арлоу., А.Нейштадт. – СПб.: Символ-Плюс, 2007. – 624 с.
2. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений / Г. Буч. – М.: Вильямс, 2008. – Часть I. – 312 с.
3. Казанская О.В. Формирование информационной образовательной среды технического университета / О.В. Казанская, В.И. Гужов // Университетское управление. – 2003. – № 4(27). – С. 57–61.
4. Крачтен Ф. Введение в Rational Unified Process / Филипп Крачтен. – М.: Вильямс, 2002. – 240 с.
5. Орлов С.А. Технологии разработки программного обеспечения: Учебник для вузов / С.А. Орлов. – СПб.: Питер, 2004. – 527 с.
6. Рамбо Дж., Блаха М. UML 2.0. Объектно-ориентированное моделирование и разработка / Дж., Рамбо, М. Блаха. – СПб.: Питер, 2016. – 713 с.