

УДК 378+004.4

Ганна Погромська,
ORCID iD 0000-0002-6779-3995
кандидат педагогічних наук, доцент,
доцент кафедри теорії й методики
природничо-математичної освіти
та інформаційних технологій
Миколаївський обласний інститут
післядипломної педагогічної освіти
вул. Адміральська, 4-а, 54001, м. Миколаїв, Україна,
hanna.pohromska@moippo.mk.ua

Наталія Махровська,
ORCID iD 0000-0001-9603-6902
кандидат фізико-математичних наук,
доцент кафедри теорії й методики
природничо-математичної освіти
та інформаційних технологій
Миколаївський обласний інститут
післядипломної педагогічної освіти
вул. Адміральська, 4-а, 54001, м. Миколаїв, Україна,
natalya.makhrovska@moippo.mk.ua

МЕТОДОЛОГІЧНІ ОСНОВИ ВИКОРИСТАННЯ МОВИ UML ДЛЯ ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

У статті розглянуто процес моделювання інформаційної системи за допомогою уніфікованої мови моделювання UML. Моделювання є центральною ланкою всієї діяльності зі створення якісного програмного забезпечення. Описано алгоритм моделювання інформаційно-освітнього середовища у спрощеному вигляді на прикладі інформаційно-освітнього середовища ЗВО із застосування мови моделювання UML. Виділено головні дійові особи (актори) з їхніми базовими функціями: викладач (створення нових курсів), студент (вибір курсу, що вивчається); менеджер (загальне управління навчальним процесом). Запропонований алгоритм є уніфікованим і може бути застосований для будь-яких інформаційних систем аналогічного типу. У дослідженні продемонстровано загальні підходи до побудови інформаційних систем у поєднанні з перевагами структурних та об'єктних методів проєктування в програмному засобі Rational Rose. Розглянуто лише умовно-спрощену модель опису інформаційно-освітнього середовища. Запропоновану модель можна розширити через описання процесів зі зв'язками між різними підрозділами середовища під час виконання конкретних практичних завдань та моделювання документів. Таке розширення має відбуватися перед створенням діаграми варіантів використання UML.

Ключові слова: діаграма; інформаційна система; інформаційно-освітнє середовище; моделювання; предметна область; проєктування; Rational Rose; UML.

© Погромська Г. С., 2020,

© Махровська Н. А., 2020

Вступ. У сучасному світі розробляється велика кількість різноманітного програмного забезпечення. Одна з основних проблем під час створення великих і складних систем, зокрема програмного забезпечення (ПЗ), – це проблема складності (тех-

нічна складність і складність управління). Моделювання є центральною ланкою всієї діяльності зі створення якісного ПЗ. Моделі будуються для того, щоб зрозуміти й осмислити структуру та поведінку майбутньої системи, полегшити управління процесом її створення і зменшити можливий ризик, а також документувати прийняті проєктні рішення.

Сучасне бачення ролі моделей у розробленні програмного забезпечення відображають рекомендації С. Амблера, які мають назву Agile Modeling (Буч Г., 2008).

На думку авторів Ф. Брукса, А. М. Вендро́ва, А. Ко́берна, С. А. Орло́ва, І. Со́ммерві́лла (Орлов С. А., 2004; Со́ммерві́лл І., 2012; Брукс Ф., 2009; Вендров А. М., 2015; Ко́берн А., 2012) особливістю процесу проєктування сучасного програмного забезпечення (ПЗ) є:

- складність – невід’ємна характеристика створюваного ПЗ;
- відсутність повних аналогів ПЗ;
- наявність успадкованого ПЗ і необхідність його інтеграції з розроблюваним ПЗ;
- територіально-розподілене та неоднорідне середовище функціонування;
- велика кількість учасників проєктування;
- роз’єднаність і різноманітність окремих груп розробників за рівнем кваліфікації та досвіду.

При цьому проєктування системи є пошуком відповіді на питання про те, як слід виконати поставлену задачу. Під час проєктування увага зосереджена насамперед на задоволенні функціональних вимог і адаптації проєкту до майбутньої реалізації.

Зазначену проблему досліджували Г. Буч, В. В. Кулямін, А. Якобсон, Дж. Рамбо та ін. (Буч Г., 2008; Кулямин В. В., 2007; Якобсон А., Буч Г., Рамбо Дж., 2002). Автори виділяють такі види діяльності у процесі проєктування програмного забезпечення:

- Проєктування архітектури системи.
 - Ідентифікація архітектурних рішень і механізмів проєктування.

- Виявлення проєктних елементів системи (класів, підсистем і інтерфейсів).
- Наповнення архітектурних рівнів проєктними елементами.
- Проєктування структури потоків управління.
- Проєктування конфігурації системи.
- Проєктування елементів системи.
 - Уточнення реалізацій варіантів використання.
 - Проєктування підсистем.
 - Проєктування класів.
 - Проєктування баз даних.

Моделювання предметної галузі є одним із найважливіших етапів під час проєктування програмних систем масштабу підприємства, виробництва, організації, середовища тощо.

Для полегшення роботи розробників та автоматизованого виконання деяких стандартних дій використовуються CASE-засоби (Computer Aided Software Engineering). На певному етапі вони забезпечують підтримку більшості процесів життєвого циклу ПЗ – CASE-технології розроблення ПЗ. CASE-технологія – це сукупність методів проєктування ПЗ та інструментальних засобів для моделювання предметної області, аналізу моделей на всіх стадіях життєвого циклу ПЗ і розроблення ПЗ.

У наш час для моделювання предметної галузі на ринку програмних продуктів представлено широкий спектр CASE-засобів. Найпопулярнішими в нашій країні CASE-засобами є Rational Rose, CA BPwin, Silverrun, Sybase PowerDesigner. Моделювання предметної галузі в цих засобах має більше подібних рис, ніж відмінностей. Однак важливим, на наш погляд, є комплексність підходу та застосування єдиної уніфікованої нотації не тільки на етапі моделювання предметної галузі, але й на наступних етапах розроблення програмної системи, як у Rational Rose (Рамбо Дж., Блаха М., 2016; Арлоу Дж., Нейштадт А., 2007).

Характерним прикладом інформаційної системи, яка найповніше відобразить усі складові проектування класичної моделі, є інформаційне освітнє середовище. Така система реалізує на сучасному рівні функції не лише освітнього процесу, але й управління ним та його якістю (від набору студентів та слухачів до маркетингу освітніх послуг і формування та реалізації освітніх програм). Основна мета інформаційно-освітнього середовища університету полягає в забезпеченні можливості віддаленого інтерактивного доступу (в авторизованому режимі, орієнтованому на різні групи користувачів) до всіх освітніх ресурсів університету (Казанская О. В., 2003).

Мета роботи – продемонструвати уніфікований із методологічного погляду підхід до проектування інформаційних систем на прикладі побудови інформаційної системи закладу.

Для досягнення мети поставлено такі задачі:

1. Спроекувати модель інформаційної системи із застосуванням уніфікованої мови моделювання UML (Unified Modeling Language) із поєднанням переваг структурних та об'єктних методів проектування в програмному засобі Rational Rose.

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

Виклад основного матеріалу. Основними задачами під час моделювання предметної галузі є опис (Marshall C., 2000; Van der Aalst W., 2016):

- процесів підприємства;
- дійових осіб процесів;
- функцій, що підлягають автоматизації відповідно до структури підприємства;
- сутностей;
- сценаріїв виконання функцій, які підлягають автоматизації;
- станів сутностей;
- правил.

Уніфікована мова моделювання UML – це мова для визначення, уявлення, про-

ектування та документування програмних систем, організаційно-економічних систем, технічних систем та інших систем різної природи. UML містить стандартний набір діаграм і нотацій найрізноманітніших видів (Офіційний сайт проекту UML; Трофимов С., 2009).

В UML є три різновиди будівельних компонент (Буч Г., Якобсон И., Рамбо Дж., 2016; Фаулер М., 2015):

- предмети,
- відношення,
- діаграми.

Предмети – це абстракції, які є основними елементами в моделі, їх пов'язують певні відношення, а діаграми групують колекції предметів.

Проектувати інформаційну систему будемо на прикладі інформаційно-освітнього середовища ЗВО. Припустимо, що інформаційно-освітнє середовище, яке нами створюється (опис інформаційно-освітнього середовища розглянемо у спрощеному вигляді), буде застосовуватися:

- викладачем – для створення нових курсів;
- студентом – для вибору курсу, що вивчається;
- менеджером – для загального управління навчальним процесом.

Проектні механізми є перехідним етапом від механізмів аналізу до механізмів реалізації. Механізм реалізації – рішення, що має конкретного постачальника, проектний механізм – каркас, що максимально наближений до реалізації, що має конкретне наповнення, чим і відрізняється від механізму аналізу, що є лише своєрідною міткою.

Моделювання інформаційно-освітнього середовища починається зі створення діаграми прецедентів (UseCase Diagram). Цей тип діаграми представлений акторами, елементами UseCase та відношеннями між ними (Орлов С. А., 2004).

Будь-який елемент UseCase є деякою частиною функціональності, реалізованої в системі. Ці елементи можна ідентифікувати під час розгляду кожної взаємодії

актора з системою. У запропонованому прикладі: актор Student намагається отримати доступ до деякого курсу (Register); актор Tutor бажає запросити інформацію про особливості курсу, що викладається (Request); актор Manager повинен управ-

ляти різними організаційними питаннями (ManageEP (Manage Educational Process)). Отже можна отримати таку діаграму прецедентів (рис. 1)

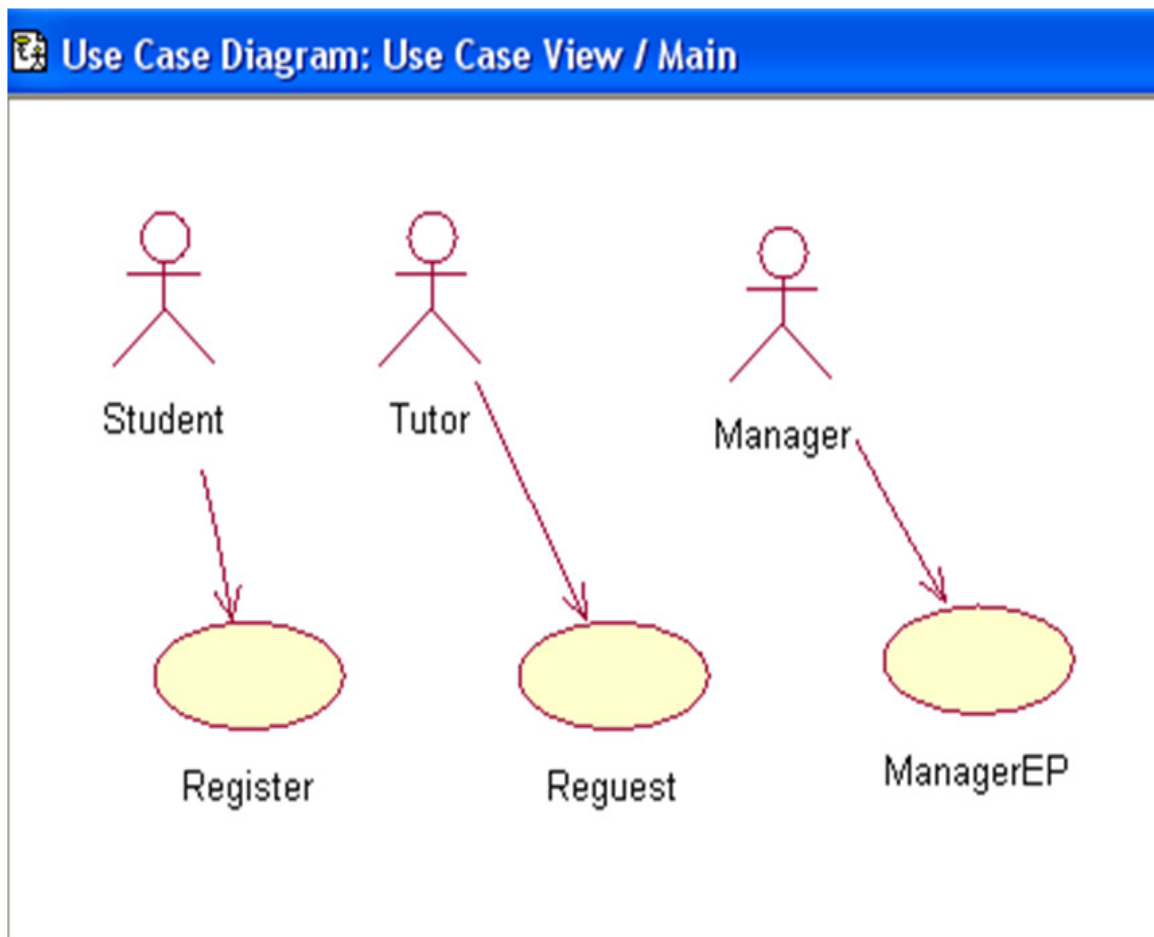


Рис. 1. Діаграма прецедентів між елементами і акторами

Джерело: складено авторами самостійно

У деяких випадках під час моделювання предметної галузі потрібно описувати сценарій роботи дійової особи з сутностями та стан сутностей.

Сценарії функцій предметної галузі можна 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 у подальшому дозволить уточнити функціональні вимоги до системи.

Докладніше функціональність елемента UseCase відображає діаграма послідовності (Sequence Diagram). Такі діаграми демонструють об'єкти та повідомлення між ними, які наочно представляють реалізацію поведінки. На рис. 2. дано приклад сценарію зарахування студента до певного курсу, описаного із застосуванням діаграми послідовності дій UML (Sequence Diagram) для елемента UseCase Register. У запропонованому сценарії студент зараховується до курсу проектування програмних систем (PPS).

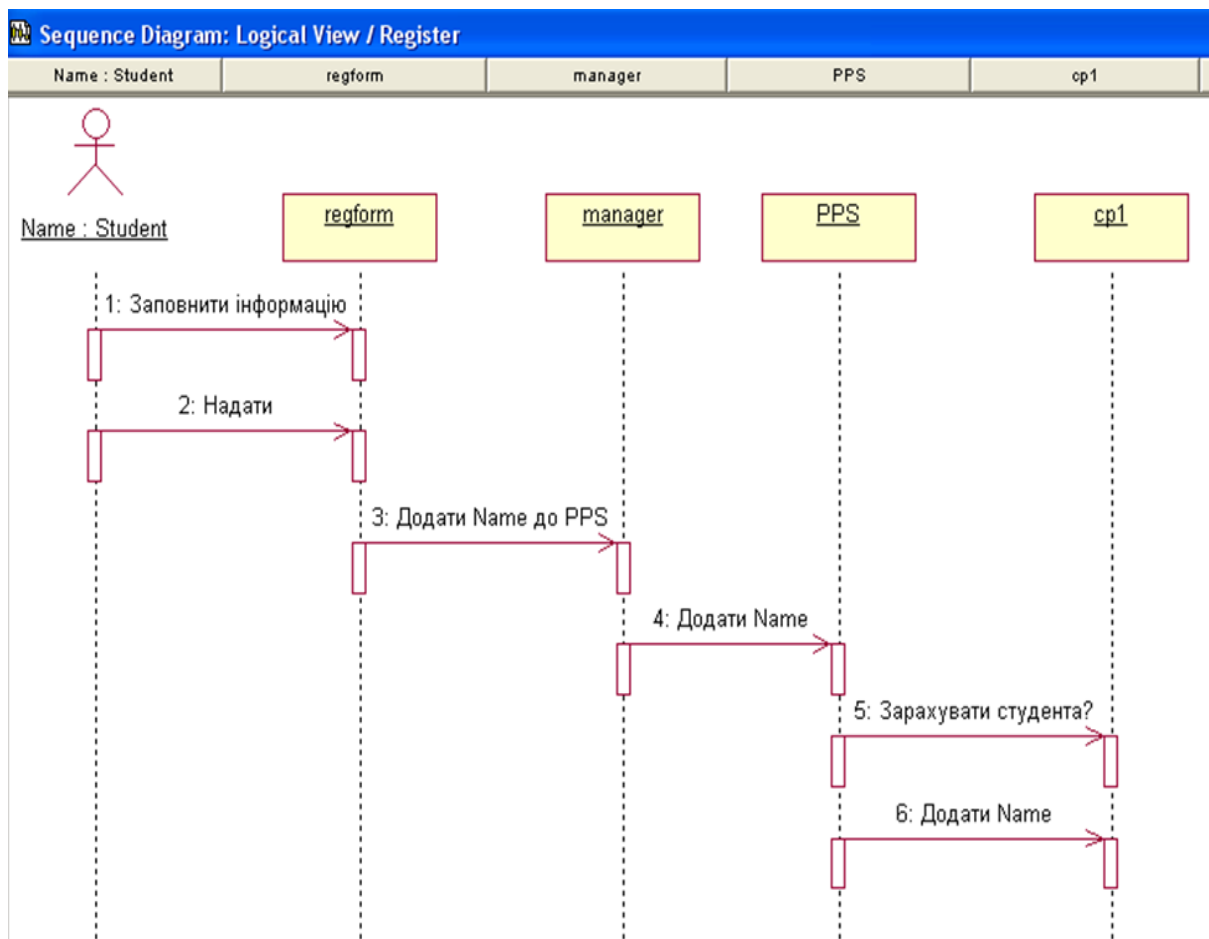


Рис. 2. Діаграма послідовності

Джерело: складено авторами самостійно

Опишемо сценарій, реалізований у діаграмі з рисунку 2. 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 такого запиту повинен бути об'єкт-курс PPS. Якщо він є, 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

указав у реєстраційній формі.

Об'єкти, представлені за допомогою діаграм послідовності, групуються в класи. Виходячи з розглянутої діаграми, можна ідентифікувати такі 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.

Важливість ролі моделей у забезпеченні життєвого циклу ПЗ дозволяє перейти до об'єктної моделі, яка є концептуальною базою об'єктно-орієнтованого підходу. Вона є збірником ідей (понять, принципів) для застосування об'єктно-орієнтованого підходу Г. Буча (Буч Г., 2008).

Побудова діаграми класів є наступним етапом у процесі опису інформаційно-освітнього середовища на мові UML (Орлов С. А., 2004).

Після того, як додані основні класи,

потрібно коректно вказати відношення між ними. Для того, щоб відобразити, як об'єкти мають взаємодіяти один з одним, вивчають діаграми послідовності та визначають шляхи взаємодії між ними.

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

Для відображення кількості об'єктів, які беруть участь у відношенні, до агрегацій та асоціацій можна додати індикатори потужності (один до одного, один до багатьох тощо).

Будь-який клас характеризується певною структурою даних та поведінкою. Наприклад, клас CoursePreparation має один атрибут numberStudents та дві операції Add() та OpenPreparation(), які описують його поведінку. Таким чином діаграма класів закономірно матиме такий вигляд (рис. 3).

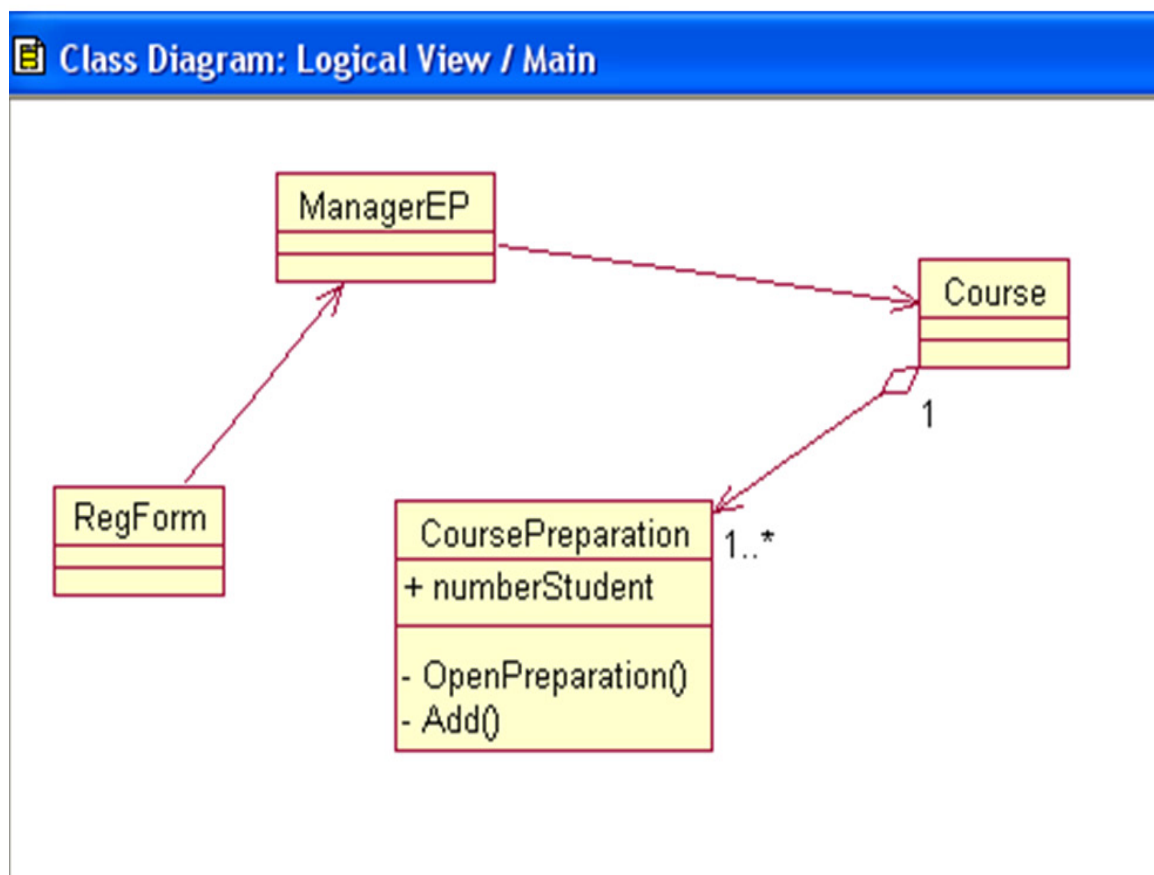


Рис. 3. Діаграма класів

Джерело: складено авторами самостійно

Описана модель наведена для демонстрації процесу проєктування 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 варто скориста-

тися 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.

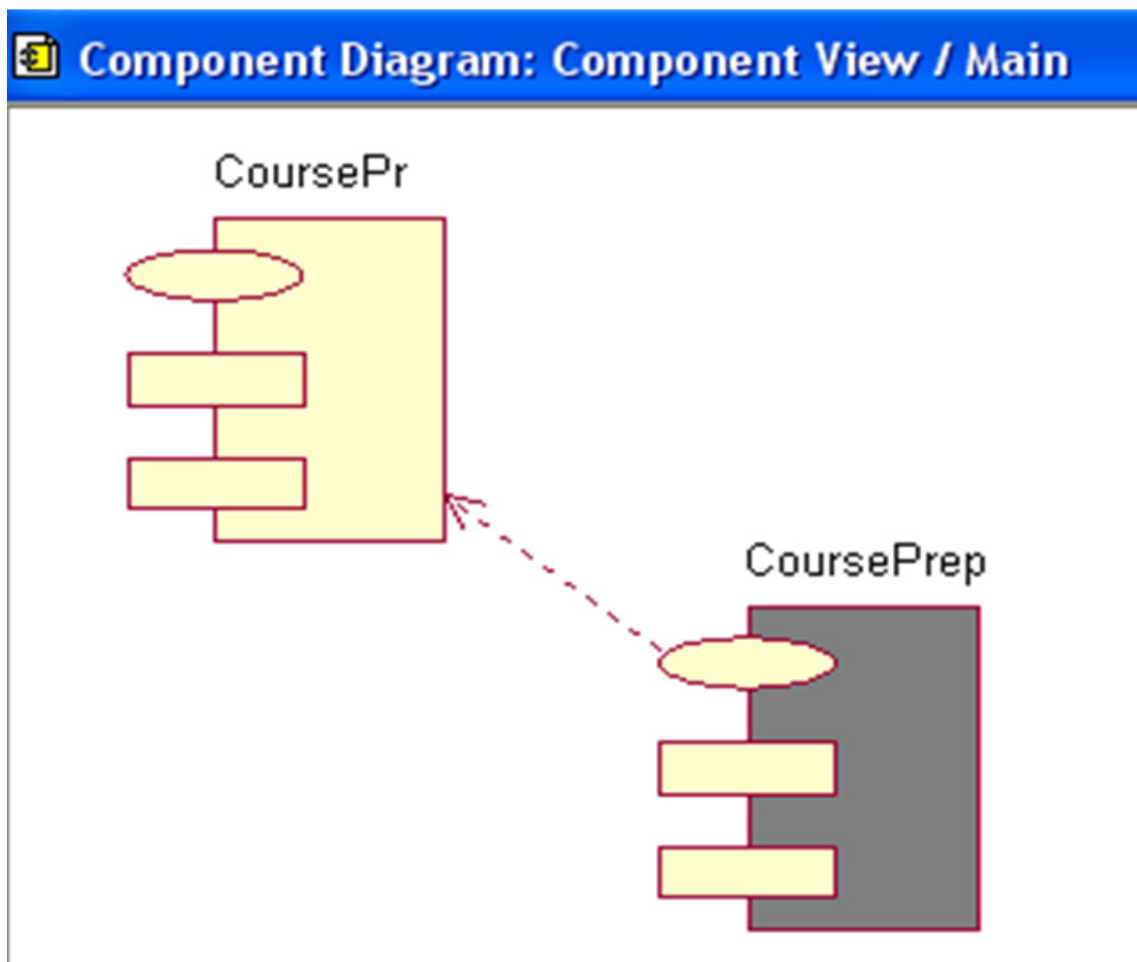


Рис. 4. Діаграма компонентів

Джерело: складено авторами самостійно

Модель інформаційно-освітнього середовища ЗВО передбачає процес моделювання документів. Мета моделювання документів – описати атрибути документів, їхні типи, значення, правила формування для проектування користувацького інтерфейсу системи, проектування бази даних системи, формування альбому вихідних форм системи (діаграми класів (Class Diagram) мови UML). Додатковою перевагою під час моделювання документів у Rational Rose є можливість приєднання до моделі документа або сутності опису його загального виду з прикладом, підготовленим зокрема в текстовому процесорі Microsoft Word.

Окрім того, результати моделювання предметної галузі в програмному CASE-засобі Rational Rose можна перетворити в документацію проекту, що відповідає промисловим стандартам розроблення програмних систем і є необхідним компонентом пакету

документації для будь-якого промислового проекту інформаційної системи. Пакетна документація підтримки процесу проектування передбачає створення версій різних моделей інформаційної системи.

Висновки. У дослідженні описано алгоритм моделювання інформаційно-освітнього середовища у спрощеному вигляді на прикладі інформаційно-освітнього середовища ЗВО із застосування мови моделювання UML. Виділено головні дійові особи (акторів) з їхніми базовими функціями: викладач (створення нових курсів), студент (вибір курсу, що вивчається); менеджер (загальне управління навчальним процесом). Автори ідентифікували такі об'єкти та класи: regform – об'єкт класу RegForm, manager – об'єкт класу ManagerEP, PPS – об'єкт класу Course, pc1 – об'єкт класу CoursePreparation. Запропонований алгоритм є уніфікованим і може бути застосо-

ваний для будь-яких інформаційних систем аналогічного типу. У дослідженні продемонстровано загальні підходи до побудови інформаційних систем із перевагами структурних та об'єктних методів проектування в програмному засобі Rational Rose.

У статті розглянута лише умовно-спрощена модель опису інформаційно-освітнього середовища. Запропоновану модель можна розширити через опис процесів зі зв'язками між різними підрозділами середовища під час виконання конкретних практичних завдань (горизонтальні зв'язки на діаграмі діяльності (Activity Diagram),

що відбувається перед створенням діаграми варіантів використання UML (UseCase Diagram), та моделювання документів (Крачтен Ф., 2002).

Опис предметної галузі із застосуванням UML добре сприймають експерти предметної галузі. Отже, представлені елементи опису інформаційно-освітньої системи на базі мови UML дозволяють зробити висновок про достатню легкість і водночас достатню гнучкість під час застосування подібних технологій для проектування сучасних інформаційно-освітніх систем.

ЛІТЕРАТУРА

1. Амблер С. Гибкие технологии: экстремальное программирование и унифицированный процесс разработки / С. Амблер. – СПб. : Питер, 2015. – 416 с.
2. Арлоу Дж. UML 2 и унифицированный процесс. Практический объектно-ориентированный анализ и проектирование / Дж. Арлоу, А. Нейштадт. – СПб. : Символ-Плюс, 2007. – 624 с.
3. Брукс Ф. Мифический человеко-месяц или как создаются программные системы / Ф. Брукс. – СПб. : Символ-Плюс, 2009.
4. Буч Г. UML. Классика CS / Г. Буч, И. Якобсон, Дж. Рамбо. – СПб. : Питер, 2016. – 736 с.
5. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений / Г. Буч. – М. : Вильямс, 2008. – Часть I. – 312 с.
6. Вендров А. М. Проектирование программного обеспечения экономических информационных систем : учебник. – М. : Финансы и статистика, 2015. – 544 с.
7. Казанская О. В. Формирование информационной образовательной среды технического университета / О. В. Казанская, В. И. Гужов // Университетское управление. – 2003. – № 4(27). – С. 57–61.
8. Коберн А. Быстрая разработка программного обеспечения / А. Коберн. – М. : ЛОРИ, 2012. – 304 с.
9. Крачтен Ф. Введение в Rational Unified Process / Филипп Крачтен. – М. : Вильямс, 2002. – 240 с.
10. Кулямин В. В. Технологии программирования. Компонентный подход / В. В. Кулямин. – М. : Бинум. Лаборатория знаний. – 2007. – 315 с.
11. Орлов С. А. Технологии разработки программного обеспечения : учебник для вузов. / С. А. Орлов. – СПб. : Питер, 2004. – 527 с.
12. Офіційний сайт проекту UML [Електронний ресурс]. – Режим доступу: <http://www.uml.org>
13. Рамбо Дж., Блаха М. UML 2.0. Объектно-ориентированное моделирование и разработка / Дж. Рамбо, М. Блаха. – СПб.: Питер, 2016. – 713 с.

14. Соммервилл И. Инженерия программного обеспечения / И. Соммервилл. – М. : Вильямс, 2012. – 425 с.
15. Трофимов С. Как научиться применять UML? / С. Трофимов [Электронный ресурс]. – Режим доступа: <http://www.caseclub.ru/articles/uml.html>
16. Фаулер М. UML. Основы. Краткое руководство по стандартному языку объектного моделирования / М. Фаулер. – СПб. : Символ-Плюс, 2015. – 192 с.
17. Якобсон А. Унифицированный процесс разработки программного обеспечения / А. Якобсон, Г. Буч, Дж. Рамбо. – СПб. : Питер, 2002. – Глава 9. – 496 с.
18. Marshal, C. Enterprise Modeling with UML. Designing Successful Software through Business Analysis / Chris Marshal. – Addison-Wesley Professional, 2000. – 259 p.
19. Van der Aalst, W. Data science in action. In Process Mining, Springer, 2016. – P. 3–23. – Retrieved from: http://dx.doi.org/10.1007/978-3-662-49851-4_1

МЕТОДОЛОГИЧЕСКИЕ ОСНОВЫ ИСПОЛЬЗОВАНИЯ ЯЗЫКА UML ДЛЯ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Погромская Анна,

*кандидат педагогических наук, доцент,
доцент кафедры теории и методики
естественно-математического образования
и информационных технологий
Николаевский областной институт
последипломного педагогического образования
ул. Адмиральская, 4-а, 54001, г. Николаев, Украина
hanna.pohromska@moipro.mk.ua*

Махровская Наталья,

*кандидат физико-математических наук,
доцент кафедры теории и методики
естественно-математического образования
и информационных технологий
Николаевский областной институт
последипломного педагогического образования
ул. Адмиральская, 4-а, 54001, г. Николаев, Украина
natalya.makhrovska@moipro.mk.ua*

В статье рассмотрен процесс моделирования информационной системы с помощью унифицированного языка моделирования UML. Моделирование является центральным звеном всей деятельности по созданию качественного программного обеспечения. Описан алгоритм моделирования информационно-образовательной среды в упрощенном виде на примере информационно-образовательной среды высшего учебного заведения с использованием языка моделирования UML. Выделены главные действующие лица (актёры) с их базовыми функциями: преподаватель (создание новых курсов), студент (выбор изучаемого курса), менеджер (общее управление учебным процессом). Предло-

женный алгоритм является унифицированным и может быть применен для любых информационных систем аналогичного типа. В исследовании продемонстрированы общие подходы к построению информационных систем в объединении с преимуществами структурных и объектных методов проектирования в программном средстве *Rational Rose*. Рассмотрена лишь условно упрощенная модель описания информационно-образовательной среды. Предложенную модель можно расширить с помощью описания процессов со связями между различными подразделениями среды при решении конкретных практических задач (горизонтальные связи на диаграмме *Activity Diagram*) и моделирования документов. Такое расширение должно происходить перед созданием диаграммы вариантов использования UML (*UseCase Diagram*).

Ключевые слова: диаграмма; информационная система; информационно-образовательная среда; моделирование; предметная область; проектирование; *Rational Rose*; UML.

METHODOLOGICAL USE THE UML LANGUAGE FOR DESIGNING AN INFORMATION SYSTEM

Pohromska Hanna,

PhD, Docent,

Associate Professor,

Department of Theory and Methods of Sciences,

Mathematics and Information Technologies

Mykolaiv In-Service Teachers Training Institute

4-a Admiralska Street, 54001, Mykolaiv, Ukraine,

hanna.pohromska@moippo.mk.ua

Makhrovska Natalya,

PhD, Associate Professor,

Department of Theory and Methods of Sciences,

Mathematics and Information Technologies

Mykolaiv In-Service Teachers Training Institute

4-a Admiralska Street, 54001, Mykolaiv, Ukraine

natalya.makhrovska@moippo.mk.ua

The article discusses the information system modeling through using the unified modeling language UML. Modeling is at the heart of quality software development. The authors describe an algorithm for modeling of the educational environment in a simplified form. As an example, the information educational environment of a higher educational institution was selected.

The main roles and their basic functions are highlighted. The first role is a teacher's one (to create and to fill educational courses), the second role is a student's one (to choose and study the course. The third role is designed to manage the educational process. The algorithm proposed by the authors is universal. It can be applied to any information systems of a similar type. This research demonstrates general approaches to building information systems. It also illustrates the benefits of structured and object design methods in the Rational Rose software.

In the considered model, for example, only a few components are described to demonstrate general approaches to design. The example builds a UseCase diagram to establish links between elements and roles. The following is an example of Sequence diagram to describe the interaction between objects and their messages. And at the final stage, a generalizing Class diagram with association and aggregation relations is built. The proposed model can be extended by adding a process description with connections between units of the environment when solving specific problems (horizontal connections on the Activity Diagram). This extension must take place before creating a UML (UseCase Diagram).

Keywords: *design, diagram, information-educational environment, information system, modeling, Rational Rose, subject area, UML.*

REFERENCES

1. Ambler, S. (2015). *Gibkie tehnologii: ekstremalnoe programmirovaniye i unifikirovannyiy protsess razrabotki* [Flexible technologies: extreme programming and unified development process]. SPb: Piter (rus).
2. Arlou, Dj., & Neyshadt, A. (2007). *UML 2 i Unifikirovannyiy protsess. Prakticheskiy obyektno-orientirovannyiy analiz i proektirovaniye* [UML 2 and Unified Process. Practical object-oriented analysis and design]. SPb.: Simvol-Plyus (rus).
3. Bruks, F. (2009). *Mificheskiy cheloveko-mesyats ili kak sozdayutsya programmnyie sistemyi* [The mythical man-month or how software systems are created]. SPb.: Simvol-Plyus (rus).
4. Buch, G. (2008). *Obyektno-orientirovannyiy analiz i proektirovaniye s primerami prilozheniy* [Object Oriented Analysis and Design with Sample Applications]. Moscow: Vilyams (rus).
5. Buch, G., Yakobson, I. & Rambo, Dj. (2016). *UML. Klassika CS* [UML. Classic CS]. SPb.: Piter (rus).
6. Fauler, M. (2015). *UML. Osnovyi. Kratkoe rukovodstvo po standartnomu yazyiku obyektного modelirovaniya* [UML. Basics. A quick guide to the standard object modeling language]. SPb: Simvol-Plyus (rus).
7. Kazanskaya, O. V. & Gujov, V. I. (2003). *Formirovaniye informatsionnoy obrazovatelnoy sredy tehnicheskogo universiteta* [Formation of the educational information environment of a technical university]. *Universitetskoe upravlenie*, 4(27), 57–61 (rus).
8. Kobern, A. (2012). *Byistraya razrabotka programmnogo obespecheniya* [Rapid software development]. Moscow: LORI (rus).
9. Krachten, F. (2002). *Vvedeniye v Rational Unified Process* [Introduction to the Rational Unified Process]. Moscow: Vilyams (rus).
10. Kulyamin, V. V. (2007). *Tehnologii programmirovaniya. Komponentnyiy podhod* [Programming technologies. Component approach]. Moscow: Binom. Laboratoriya znaniy (rus).
11. Marshal, C. (2000). *Enterprise Modeling with UML. Designing Successful Software through Business Analysis*. Addison-Wesley Professional (eng).
12. Orlov, S. A. (2004). *Tehnologii razrabotki programmnogo obespecheniya* [Software development technologies]. SPb.: Piter (rus).

13. Official site of the UML project [Electronic resource]. Retrieved from: <http://www.uml.org> (eng).
14. Rambo, Dj. & Blaha, M. (2016). *UML 2.0. Obyektno-orientirovannoe modelirovanie i razrabotka* [UML 2.0. Object-oriented modeling and development]. SPb.: Piter (rus).
15. Sommervill, I. (2012). *Injeneriya programmnogo obespecheniya* [Software engineering]. Moscow: Vilyams (rus).
16. Trofimov, S. Kak nauchitsya primenyat UML? [How to learn to apply UML?] [Electronic resource]. Retrieved from: <http://www.caseclub.ru/articles/uml.html> (rus).
17. Van der Aalst, W. (2016). Data science in action. In *Process Mining*, [Electronic resource]. Retrieved from : http://dx.doi.org/10.1007/978-3-662-49851-4_1 (eng).
18. Vendrov, A. M. (2015). *Proektirovanie programmnogo obespecheniya ekonomicheskikh informatsionnyih sistem* [Software design of economic information systems]. Moscow: Finansyi i statistika (rus).
19. Yakobson, A., Buch, G. & Rambo, Dj. (2002). *Unifitsirovannyiy protsess razrabotki programmnogo obespecheniya* [Unified software development process]. Glava 9. SPb.: Piter (rus).