

Міністерство освіти та науки, молоді та спорту України
Криворізький національний університет
Національний педагогічний університет
імені М. П. Драгоманова
Харківський національний педагогічний університет
імені Г. С. Сковороди
Черкаський національний університет
імені Богдана Хмельницького

Комп'ютерне моделювання в освіті

*Матеріали V Всеукраїнського
науково-методичного семінару*

6 квітня 2012 року

Кривий Ріг 2012

ОБ'ЄКТНО-ОРІЄНТОВАНЕ МОДЕЛЮВАННЯ ЯК ТЕХНОЛОГІЯ ДОСЛІДЖЕННЯ СКЛАДНИХ СИСТЕМ ТА ЯК НАВЧАЛЬНА ДИСЦИПЛІНА

О. І. Теплицький

м. Кривий Ріг, Криворізький національний університет

Прискорення розвитку технологій (особливо інформаційно-комунікаційних) створює утруднення для системи освіти: упровадження засобів ІКТ в методичні системи навчання змінює всі компоненти системи, у тому числі – зміст навчання. Для інформатичних дисциплін вплив зміни засобів навчання на зміну змісту навчання є суттєво більшим, ніж для інших дисциплін, що породжує проблему усталеності змісту навчання. Розв'язання цієї проблеми можливе за двома основними напрямками: 1) усталення змісту навчання через ігнорування впливу зміни технологій; 2) усталення змісту навчання через усталення технологій. При цьому жоден з них у повній мірі не розв'язує проблему.

Перспективним напрямом вбачається фундаменталізація навчання інформатичних дисциплін через посилення ролі основного методу дослідження в інформатиці – методу моделювання, що одночасно виступає в якості провідного універсального методу навчання. При цьому урахування психологічних особливостей відображення свідомістю людини об'єктів оточуючої дійсності вимагає відповідної їх інтерпретації у комп'ютерних моделях. Таким чином, розв'язання проблеми фундаменталізації змісту навчання інформатичних дисциплін в умовах швидкої зміни засобів ІКТ потребує об'єднання методу моделювання та об'єктно-орієнтованої технології програмування, що разом утворюють якісно нову концепцію – *об'єктно-орієнтоване моделювання*.

Об'єктно-орієнтоване програмування (ООП), що отримало широке розповсюдження як потужна технологія програмування, містить у собі традиційні процедурні методи структурного програмування, а об'єктний підхід включає підхід структурний. Популярність ООП у чималій мірі визначається більш сильною формою структуризації програмного забезпечення (ПЗ), що створюється на його основі. Використання ООП прискорює процес розробки програм, надаючи можливість швидкої модифікації створюваного ПЗ. Підґрунтям ООП є більш загальна технологія пізнання, що має назву об'єктного підходу до відображення дійсності. За об'єктного підходу реальні об'єкти відображаються у свідомості в об'єктно-орієнтованій моделі.

Основне утруднення, що постає перед розробником складного програмного продукту, полягає в неспроможності одночасно утримувати в

пам'яті усі необхідні деталі. Г. Р. Міллер та його послідовники стверджують (див., наприклад, [1]), що максимальна кількість об'єктів, з якою здатен одночасно оперувати людський мозок, не перевищує 7 ± 2 (так званий «гаманець Міллера»). Це «магічне число», скоріше за все, пов'язане з обсягом короткострокової пам'яті у людини. Ще одним обмежуючим фактором тут виступає швидкість опрацювання мозком нової інформації: йому потрібно приблизно 5 секунд на сприймання кожного нового об'єкту. Як бачимо, природна здатність людського мозку до роботи зі складними системами є низькою.

Проте, як услід за Е. Дейкстрою зазначає Б. Страуструп [2], ще з давніх часів людству відомий простий та ефективний спосіб управління складними системами: «Розділяй та володарюй». Тому при проектуванні складної системи слід складати її з окремих невеликих підсистем – у цьому випадку ми не виходимо за межі можливостей людини: при розробці системи будь-якого рівня складності необхідно одночасно утримувати в пам'яті інформацію лише про деякі її частини.

Такий підхід називається *алгоритмічною декомпозицією* і забезпечує психологічне підґрунтя для процедурного програмування, визначаючи головну вимогу до написання підпрограми: «усі дії, що виконуються в підпрограмі, повинні усвідомлюватися *одночасно*», і якщо ця вимога не виконується, підпрограму слід поділити на дрібніші блоки.

Проте число подій, що їх одночасно може опрацювати людина, *не залежить від обсягу інформації*, яка міститься у кожній події, і це дає людині надзвичайно ефективний механізм опрацювання складних повідомлень – *абстрагування*. Не маючи можливості відтворити у всіх деталях складний об'єкт, ми ігноруємо несуттєві для нас деталі і, таким чином, маємо справу з узагальненою, ідеалізованою моделлю об'єкта. І, хоча при цьому ми, як і раніше, змушені охоплювати одночасно значну кількість властивостей об'єкту, та завдяки абстрагуванню ми використовуємо узагальнені властивості суттєво більшого семантичного обсягу. Це особливо доцільно, коли ми розглядаємо світ з позицій об'єктно-орієнтованої взаємодії, оскільки об'єкти як абстракції реального світу являють собою насичені зв'язні інформаційні одиниці. При цьому ми також обмежені кількістю об'єктів, яку можемо сприйняти у кожний окремий момент. Все одно, використовуючи абстрактні поняття, ми отримуємо можливість працювати із складними системами, а, отже, і створювати складні програмні продукти. У процесі моделювання абстрагування (спрощення моделі через накладання певних припущень стосовно природи модельованого об'єкту) відбувається на етапі формалізації моделі. При цьому виділяються суттєві властивості об'єктів та відношення між ними.

Складні системи можна досліджувати, концентруючи основну увагу або на об'єктах, що фігурують у системі, або на процесах, що протікають в ній. Проте доцільніше розглядати систему як впорядковану сукупність об'єктів, які в процесі взаємодії один з одним забезпечують функціонування системи як єдиного цілого. Об'єкти, що складають систему, можуть утворювати *ієрархії*. За такого підходу основним способом дослідження складної системи є *об'єктна декомпозиція*, яка в процесі моделювання реалізується як через ієрархії наслідування об'єктів, так і через ієрархії моделей (за принципом спрощення/ускладнення).

Таким чином, з'являється можливість розширити межі пізнавальних можливостей людини, використовуючи методи декомпозиції, виділення абстракцій та створення ієрархій. Саме ці методи покладено в основу об'єктного підходу, реалізацією якого до дослідження складних систем є *об'єктно-орієнтоване моделювання* – це вид комп'ютерного моделювання, за якого середовищем моделювання є деяке середовище програмування, що надає можливість конструювання об'єктів, їх використання та обміну повідомленнями між ними.

Вид програмування, що використовується у такому середовищі, називають *об'єктно-орієнтованим програмуванням* – це технологія програмування, заснована на поданні програми у вигляді сукупності об'єктів, кожен з яких є реалізацією деякого класу, а класи утворюють ієрархію за принципами наслідуваності.

Таким чином, об'єктно-орієнтоване моделювання є видом комп'ютерного моделювання, яке, враховуючи особливості перебігу психічних процесів та відображення дійсності, надає можливість конструювати чітко структуровані та компактні комп'ютерні моделі.

Об'єктний підхід відомий ще з давніх часів. Так, давнім грекам належить ідея про те, що світ можна розглядати як у термінах об'єктів, так і подій. А в XVII ст. Р. Декарт підкреслював, що люди зазвичай мають погляд на світ, який сьогодні міг би називатися об'єктно-орієнтованим. У 60–70-х р.р. XX ст. ця думка була розвинена в одній з течій когнітивної філософії – об'єктивістській епістемології, а на початку 1980-х р.р. М. Мінські запропонував модель людського мислення, в якій розум людини розглядається як спільнота агентів, що по-різному мислять. На його думку, лише спільні дії таких агентів приводять людину до осмисленої поведінки. Реалізація таких моделей можлива в мультиагентних середовищах об'єктно-орієнтованого моделювання, таких як NetLogo, з використанням ще однієї концепції ООП – *паралелізму*, що знайшла своє втілення в архітектурі сучасних комп'ютерних систем.

Основні ідеї об'єктного підходу (абстрагування, типізація, ієрархія тощо) в тому чи іншому вигляді були присутні в практиці програмуван-

ня, починаючи з перших мов високого рівня.

Об'єктно-орієнтоване проектування – це вид проектування, що поєднує в собі процес об'єктної декомпозиції та прийоми подання логічної та фізичної, а також статичної та динамічної моделей проектованої системи [3]. Об'єктно-орієнтоване проектування відображає процес конструювання об'єктно-орієнтованої моделі.

Саме об'єктно-орієнтована декомпозиція відрізняє об'єктно-орієнтоване проектування від структурного; у першому випадку логічна структура моделі відображається абстракціями у вигляді класів і об'єктів, у другому – алгоритмами. Об'єктно-орієнтований аналіз спрямований на створення об'єктно-орієнтованих моделей реальної дійсності.

Об'єктно-орієнтований аналіз – це вид аналізу, за якого вимоги до моделі системи сприймаються з точки зору класів та об'єктів, виявлених у предметній галузі [4].

Основними етапами об'єктно-орієнтованого аналізу є:

1) *розробка інформаційної моделі*. На цьому етапі центральним є абстрагування концептуальної суті в завданні в термінах об'єктів і атрибутів. Стосунки між сутностями формалізуються у зв'язках, які ґрунтуються на лініях поведінки, правилах і фізичних законах, що превалюють у реальному світі;

2) *розробка моделі станів*. Цей етап методу пов'язаний з поведінкою об'єктів і зв'язків у часі. В об'єктно-орієнтованому аналізі кожен об'єкт і зв'язок має життєвий цикл – регулярну складову частину динамічної поведінки. Часто використовують моделі станів для формалізації життєвих циклів об'єктів і зв'язків. Моделі станів, що виражаються в діаграмах і таблицях переходу, взаємодіють між собою за допомогою подій, їх організують в ієрархію, щоб зробити систему взаємодії впорядкованою і зрозумілою;

3) *розробка моделі процесів*. На цьому етапі об'єктно-орієнтованого аналізу визначаються дії, що їх необхідно виконати для переходу об'єкту з одного стану у інший. Отримувані в такий спосіб процеси надалі можуть бути перетворені безпосередньо на оператори (методи) об'єктно-орієнтованого проектування.

Між всіма цими визначеннями існує тісний взаємозв'язок: на результатах об'єктно-орієнтованого аналізу формуються моделі, на яких ґрунтується об'єктно-орієнтоване проектування; у свою чергу, об'єктно-орієнтоване проектування створює фундамент для реалізації об'єктно-орієнтованої моделі.

Великого значення набуває розгляд сфер застосування ООП і предметних областей, де воно може виявитися найбільш ефективним, тобто

мати явні переваги в порівнянні, наприклад, з процедурним підходом. Принципова відмінність двох підходів визначається тим, що є первинним – об'єкт чи процес. У [5, 15–16] ці критерії уточнюються: якщо ідейним стрижнем програми є процес, навряд чи вживання ООП може принести якусь користь і підвищити наочність коду, ефективність розробки і супроводу. Проте існують такі розділи прикладного ПЗ, де висока ефективність і наочність вживання засобів ООП не викликає сумнівів, наприклад, імітаційне моделювання складних систем.

Застосування ООП дозволяє побудувати ефективну і гранично узагальнену схему імітаційного моделювання складних систем. У цій прикладній області об'єктний підхід повною мірою виявляє ті свої переваги, заради досягнення яких він, власне, і був розроблений: природність і наочність процесу програмування, первинність даних по відношенню до процесів і головне – легкість супроводу і подальшої модифікації моделюючої програми, якщо система зазнає зміни. У одній з перших робіт [6], де було дано формальне визначення поняття «об'єктно-орієнтоване моделювання», сказано: «Коли говорять про об'єктно-орієнтовану концепцію опису систем, то мають на увазі ПЗ у вигляді сукупності дискретних об'єктів, що містять в собі і структуру даних, і реалізацію поведінки. З іншого боку, імітаційну модель можна розглядати як деяку множину сутностей, що взаємодіють одна з одною. З цієї точки зору можна навіть сказати, що імітаційна модель завжди об'єктно-орієнтована, оскільки об'єктна парадигма – це цілком природний спосіб моделювання, якщо відображувати реальні сутності на об'єкти».

Саме це стало однією з причин включення блоку імітаційних моделей до змісту навчального курсу об'єктно-орієнтованого моделювання.

Природний зв'язок ООП та моделювання був відображений вже у першій мові ООП – Simula 67, сама назва якої походить від слова *simulation* – моделювання. Автори Simula 67 запропонували використовувати спеціальні моделі – класи, які описують множину близьких за своїми властивостями об'єктів, що мають внутрішню структуру і поведінку, та надають можливість вибирати конкретний елемент цієї множини, створюючи конкретний екземпляр класу – об'єкт, і наділяючи його конкретними значеннями параметрів.

Сучасний стан розвитку об'єктного підходу як технології опису і проектування складних багатокомпонентних систем визначає *уніфікована мова моделювання UML (Unified Modeling Language)* [7]. Модифікації об'єктно-орієнтованої технології стосовно моделювання складних багатокомпонентних динамічних систем в UML також називаються *об'єктно-орієнтованим моделюванням (ООМ)*. За допомогою ООМ зручно розв'язувати багато типових завдань моделювання, а саме: створюю-

вати бібліотеки типових компонентів як бібліотеки класів; повторно використовувати компоненти, при необхідності здійснюючи їх спеціалізацію за допомогою наслідування класів; природним чином будувати моделі з безліччю однотипних об'єктів; здійснювати параметризацію моделей за допомогою поліморфізму; при моделюванні систем із змінним складом створювати і знищувати екземпляри об'єктів в ході обчислювального експерименту.

Спрямування UML на моделювання складних багатокomпонентних динамічних систем визначило включення блоку динамічних моделей до змісту навчального курсу об'єктно-орієнтованого моделювання.

Таким чином, проведений аналіз принципів застосування об'єктно-орієнтованого підходу до моделювання надав можливість визначити новий вид комп'ютерного моделювання – об'єктно-орієнтоване моделювання, а дослідження змісту поняття об'єктно-орієнтованого моделювання надає можливість визначити його як навчальну дисципліну у такий спосіб: ***об'єктно-орієнтоване моделювання – це навчальна дисципліна, в якій вивчаються способи конструювання та дослідження об'єктно-орієнтованих моделей [8].***

Література

1. Miller G. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information / Miller G. // The Psychological Review. – 1956. – Vol. 63 (2). – P. 86.
2. Stroustrup B. What is object-oriented programming? / Stroustrup B. // IEEE Software. – 1988. – Vol. 5. – P. 10–20.
3. Meyer B. Reusability: the case of object-oriented design / Meyer B. // IEEE Software. – 1987. – Vol. 3. – P. 50–64.
4. Токарь С. В. Объектно-ориентированный анализ для программистов / С. Токарь, В. Штонда. // Soft Review. – 1993. – Октябрь. – С. 3–8.
5. Труб И. И. Объектно-ориентированное моделирование на C++ / Илья Труб. – М. [и др.] : Питер, 2006. – 410 с. – (Учебный курс).
6. Copstein B. The Object Oriented Approach and the Event Discrete Simulation Paradigms / Copstein B., Pereira C. E., Wagner F. R. // Proceedings Society for Computer Simulation. – 10th European Simulation Multiconference. – Budapest, Hungary, 1996. – June. – P. 57–61.
7. Рамбо Дж. UML 2.0. Объектно-ориентированное моделирование и разработка / Дж. Рамбо, М. Блаха. – СПб. : Питер, 2007. – 544 с.
8. Теплицький О. І. Об'єктно-орієнтоване моделювання в Alice. Частина 1 / О. І. Теплицький ; за науковою редакцією академіка НАПН України М. І. Жалдака. – К. : НПУ імені М. П. Драгоманова, 2011. – 56 с.