

УДК 004.4

В.І. Межуєв

Одеський національний політехнічний університет, Одеса

ДОЦІЛЬНІСТЬ ЗАСТОСУВАННЯ UML ДЛЯ МОДЕЛЮВАННЯ СИСТЕМ ОЗБРОЄННЯ ТА ВІЙСЬКОВОЇ ТЕХНІКИ

Предметом розгляду статті є уніфікована мова моделювання UML (Unified Modeling Language), зокрема доцільність її застосування для проектування систем озброєння та військової техніки. Дана загальна характеристика UML, проаналізовані її переваги та вади, визначені шляхи подолання недоліків «універсального» підходу до моделювання предметних областей.

Ключові слова: уніфікована мова моделювання UML; предметно-орієнтоване моделювання.

Вступ

Предметом розгляду у цій статті є методологія моделювання предметних областей (ПрО), що отримала широке поширення у сучасній системній інженерії (зокрема, при проектуванні систем озброєння та військової техніки). Мова йде про так звані універсальні (чи уніфіковані) мови, що здійснюють моделювання у системі деяких загальних для множини модельованих ПрО понять. Найбільш відомою та поширеною наразі є UML (Unified Modeling Language) - уніфікована мова моделювання, призначена для моделювання ПрО з метою розробки відповідного програмного забезпечення [1 – 11]. Зазначимо, що існують також й інші підходи до уніфікації моделювання, зокрема відзначимо методологію раціональних уніфікованих процесів RUP (Rational Unified Process), що розроблюється компанією Rational Software [11].

Постановка проблеми. Разом із багатьма перевагами, що надають уніфіковані мови моделювання предметних областей, у той же час вони страждають від великої кількості суттєвих вад. Саме тому є необхідним аналіз переваг та недоліків існуючих універсальних мов з метою визначення їх істинної застосовності до моделювання ПрО. Такий аналіз повинен не лише акцентувати увагу на вадах існуючих підходів, але давати конкретні рекомендації щодо подолання цих недоліків. У цій статті акцент робиться на дослідженні уніфікованої мови моделювання UML.

Основна частина

Історія UML сходить до 1994 року, коли Граді Буч (Grady Booch) і Джеймс Рамбо (James Rumbaugh), що працювали у Rational Software [11], об'єднали свої зусилля для створення нової мови об'єктно-орієнтованого моделювання. За основу ними були взяті власні методи, а саме орієнтований на аналіз метод Буча (Booch Method) і метод Рамбо ОМТ (Object-Modeling Technique) [6], орієнтований на проектування програмних систем.

Третім автором UML вважається Івар Якобсон (Ivar Jacobson), автор методу OOSE (Object-Oriented Software Engineering) [6].

Метод Якобсона забезпечував можливості для специфікації бізнес-процесів й аналізу вимог за допомогою сценаріїв використання (Use Case).

У результаті спільної роботи та за підтримки консорціуму OMG (Object Management Group) [12] у січні 1997 року з'явилася специфікація UML 1.0. Останньою базовою специфікацією UML вважається 2.0, що була опублікована в серпні 2005 року. У версії 2.0 семантика мови UML була значно уточнена й розширена для підтримки методології «архітектури на базі моделей» MDA (Model Driven Architecture) [13].

Остання (на час написання цієї роботи) версія UML - 2.3, була опублікована консорціумом OMG у травні 2010 року.

Але відзначимо, що реліз UML 2.0 не виправдав очікування UML-співтовариства (хоча прихильники UML 2.0 і стверджують, що ця мова відбиває передовий досвід і кращі методи моделювання ПрО). Існує і точка зору, що ми досі перебуваємо у «дитячому віці розробки мов моделювання» [14].

Загальна характеристика UML

UML здійснює декомпозицію моделей предметних областей на складові частини за допомогою подань:

- *Статичне* подання описує структурні аспекти системи (наприклад, модель класів).
- Подання *взаємодії* використовується для моделювання послідовностей дій об'єктів ПрО.
- Подання *діяльності* використовується для створення моделей процесів в системі.
- Подання у вигляді *кінцевого автомата* використовується для опису поведінки системи в термінах станів і переходів між ними.

Таким чином, подання UML надають чотири можливих семантики для моделювання ПрО. Але помітимо, що ці подання не є ортогональними: концепції, що використовуються в одному з них,

часто залежать від концепцій, застосовуваних в іншому.

Так, класифікатори учасників взаємодії повинні бути визначені в статичній структурній моделі.

Найважливішою складовою UML є графічна нотація (рис. 1), що дозволяє специфікувати, проектувати, візуалізувати і документувати артефакти об'єктно-орієнтованого програмного забезпечення [1 – 11].

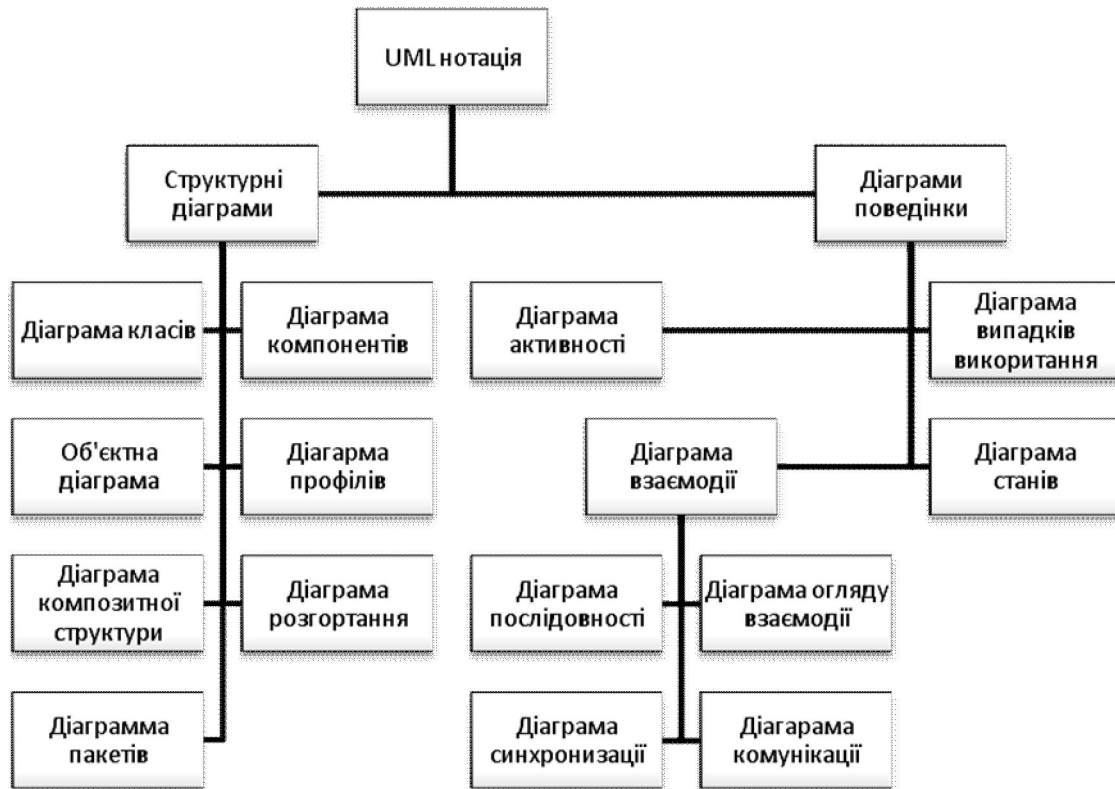


Рис. 1. Структура графічної нотації мови UML версії 2.3

Зазначимо, що графічна нотація UML спеціально розроблювалася для того, щоб діаграми було легко створювати на папері. У той же час наразі розроблені десятки CASE (Computer Aided Software Engineering) засобів, призначених для полегшення побудови діаграм у UML нотації. Серед найбільш відомих CASE засобів, що підтримують UML, зазначимо IBM Rational Rose; Microsoft Visio; Sparx Systems Enterprise Architect; Gentleware Poseidon; SmartDraw; Telelogic TAU G2; StarUML та ін. (гарний огляд CASE засобів наданий О.В.Бабіч у його введенні у UML [15]).

Зазначимо, що мета використання UML не обмежується лише розробкою програмного забезпечення. Її також застосовують для створення схем даних (Entity-Relationship Diagram, ERD), моделювання бізнес-процесів (workflows), проектування компонентів систем та ін. Помітимо, що для моделювання систем було розроблене спеціальне розширення UML - SysML (Systems Modeling Language) - мова моделювання систем [2, 16].

SysML підтримує визначення, аналіз, проектування, перевірку й підтвердження відповідності широкого спектра систем. У порівнянні з UML, що

орієнтована на моделювання програмних продуктів, SysML надає системному інженеру деякі додаткові можливості.

SysML розширює UML за рахунок введення двох додаткових типів діаграм: діаграми вимог і параметричної діаграми. Перша служить для збору та ієрархічного структурування вимог, а друга для кількісного аналізу параметрів систем (наприклад, продуктивності або надійності). У результаті розширюється коло модельованих систем, елементами яких можуть бути програмні компоненти, апаратне забезпечення, обладнання, процеси, інформація, персонал та ін.

Зазначимо, що SysML є також більш компактною нотацією, вона не включає багато програмно-орієнтованих особливостей UML, саме тому її легше вивчати й застосовувати.

Відзначимо деякі інші відомі розширення (профілі) UML – Modeling and Analysis of Real-Time Embedded systems (MARTE), Service oriented architecture Modeling Language (SoaML), UML Profile for CORBA, UML Profile for Data Distribution, UML Profile for Enterprise Application Integration (EAI), UML Profile for System on a Chip (SoC) [17] та ін.

Переваги UML

Відзначимо переваги UML, які найчастіше виділяють її розробники [1 – 11]:

Повнота опису ПрО. UML дозволяє описувати ПрО практично із всіх можливих точок зору (як структурної, так і поведінкової - див. верхній рівень діаграми на рис. 1). Відзначається, що UML 2.0 надає більшу виразність засобів моделювання (у тому числі бізнес-процесів, архітектур розподілених гетерогенних систем і т.ін.).

Розширюваність. UML дозволяє вводити власні текстові й графічні стереотипи, що сприяє розширенню області його застосування (і, таким чином, використанню не тільки в сфері програмної інженерії). У версії UML 2.0. полегшене додавання «семантичних варіацій» (введеної користувачем семантики), що приводить до визначення UML, як сімейства мов моделювання.

Уніфікованість. Завдяки своїй «універсальності» UML дозволяє розроблювачам програмного забезпечення досягти угоди в графічних позначеннях і тлумаченні семантики множини загальних понять моделювання й програмування (таких як клас, компонент, узагальнення, агрегація, поведінка та ін.).

Відповідність передовим методологіям програмування. UML є об'єктно-орієнтованою мовою моделювання, тому UML методи аналізу й проектування близькі до методів програмування у найбільш ефективній на сьогоднішній день парадигмі об'єктно-орієнтованих мов.

Незалежність від інструментів реалізації. UML є *специфікацією* мови моделювання, та не залежить від інструментів її реалізації, конкретних мов програмування й процесів розробки,

Простота графічної нотації. Діаграми UML є простими для розуміння користувачів (фахівців ПрО), після досить швидкого ознайомлення з їх синтаксисом.

Підтримка співтовариства. UML одержав широке поширення й динамічно розвивається завдяки зусиллям OMG-консорціуму [12]. Саме тому в UML здійснюється інтеграція кращого теоретичного та практичного досвіду моделювання ПрО.

Недоліки UML

Розглянемо також великий список недоліків UML [9, 14, 18] (цікавим є те, що деякі з них були також подані у списку переваг):

Множина діалектів. UML підхід претендує на універсальність, тоді як аспекти конкретних систем залишаються поза її поля зору. Для підтримки конкретної ПрО (або ж класу ПрО) розробляються окремі діалекти «універсальної» мови UML, що слугують для заповнення виникаючих «семантичних прогалів» (неповний список таких профілів був наданий нами у попередній частині статті). Зазначимо,

що врешті-решт, ці діалекти підривають повноцінність вихідної методології UML (а саме, застосовність до моделювання будь-яких ПрО).

Уніфікованість. Як слушно зазначається у [18] уніфікація є корисною далеко не завжди. Вона є доцільною на самому початку будь-якого проекту, коли фахівці з різних організацій (замовника й виконавця, підрядника й субпідрядника) повинні знайти загальну мову, щоб донести свої думки один до одного. У той же час абстракції, до яких доводиться прибгати при використанні будь-якого універсального підходу, ховають у собі багато небезпек, оскільки розуміння того, що конкретно означає кожна абстракція, у кожної людини є власним. Коли стає зрозуміло, що за систему ви збираєтеся розробляти, від універсальної мови краще перейти до мови предметної області, а не приховувати за абстракціями UML сутності ПрО.

Надмірність (перевантаженість) мови. UML включає множину практично невикористовуваних для моделювання програмних систем діаграм і конструкцій. Ця вада була ще більш збільшена в UML 2.0, тому що цей стандарт включає ще більше число «компромісів» різних фахівців співтовариства теоретиків та користувачів UML.

Складність концепцій. Стандарт UML 2.0 містить великий набір концепцій моделювання, що складним чином пов'язані між собою. Це зумовлено тим, що мова призначена для підтримки моделювання в самих різних предметних областях.

Відсутність ортогональності концепцій. UML не має строгої формальної основи, а також спирається на базові поняття, що семантично перекриваються. Інакше кажучи, UML бракує ортогональності у визначенні опорних концепцій для моделювання ПрО.

Неточна семантика. UML визначена через абстрактний синтаксис, мову опису обмежень OCL (Object Constraint Language) та природну англійську мову, специфікації в яких часто суперечать одна одній. Як наслідок, завдяки різному тлумаченню специфікацій, побудовані UML моделі виявляються несумісними в різних CASE інструментах. Крім того, не дивлячись на широке цитування, сама UML специфікація є прикладом тексту, написаного людьми, що, м'яко кажучи, не є природними носіями англійської мови.

Обмеження користувача. Як і будь-яка система позначень, UML може виразити властивості одних систем більш ефективно, ніж інші. Як наслідок, розроблювач схиляється до рішень, в яких в моделях здійснюється відхід від «чистого» UML і переплітаються сильні сторони UML і мов розробки.

UML можна визначити як не що інше, як графічний спосіб подання вимог і специфікацій, що спочатку мали текстову форму. Такий підхід змушує

розроблювачів думати в контексті пропонованих графічних нотацій і шаблонів розробки, та має результатом неоптимальні системні рішення.

Залежність від об'єктно-орієнтованої парадигми. Хоча UML створювався з метою забезпечення незалежності від конкретних мов програмування й процесів розробки, його моделі досить складно трансформуються в інші, ніж ОО, мови розробки.

Складність стикування з мовами розробки. UML визначається як мова моделювання загального призначення, що повинна бути сумісною з усіма можливими мовами розробки. Але це є практично нездійсненним завданням за перерахованими вище причинами (недостатня формалізованість, переваженість, залежність від ОО й ін.).

Складність розширень. UML 2.0 не має механізмів створення моделей з використанням понять, що відрізняються від наявних у мові. Іншими словами, опис систем з погляду та у поняттях користувача (фахівців ПрО) є неможливим.

Проблеми при вивченні й впровадженні. Вищеописані проблеми роблять проблематичним вивчення й впровадження UML. На практиці, люди (фахівці ПрО) часто будують UML діаграми з використанням символів, що надають їх CASE інструменти, не розуміючи, що насправді ці символи позначають в UML специфікації.

Неможливість зміни процесу моделювання. Хоча UML повинна бути застосовна у контексті будь-якого процесу моделювання, досвід доводить, що для різних організацій і ПрО необхідне визначення власних процесів. Наприклад, процес розробки програмного забезпечення для систем реального часу значно відрізняється від розробки програмного забезпечення для інших цілей.

Ефективність коду. Цінність UML полягає в можливості генерації з моделі ефективного вихідного програмного коду або ж коду, що можна виконати. Однак UML-модель не має властивостей повноти за Тьюрингом і генерований код в основному обмежується структурою класів й оголошенням функцій.

Шляхи подолання недоліків

Визначимо шляхи подолання недоліків універсальної мови моделювання UML, які були визначені нами раніше.

Множина діалектів та універсальність. Досвід доводить, що неможливо побудувати мову чи інструмент, пригідний для моделювання всіх без винятку ПрО. Саме тому кожна ПрО повинна мати власну мову моделювання, призначену для розв'язання специфічних для ПрО задач. Це приводить нас до розуміння та необхідності визначення принципів інструментарію, що дозволяє розроблювати інші, (предметно-орієнтовані) мови моделювання.

Надмірність (перевантаженість) мови та складність концепцій. Користувач повинен сам вибирати систему понять і зв'язків між ними, необхідних і достатніх для моделювання ПрО. Саме тому такий підхід визначається як *предметно-орієнтований*. Це дозволяє позбавитися як переваженості так і складності мови моделювання. Користувач може створити та надалі використовувати при моделюванні ПрО лише ті поняття, семантику яких він добре розуміє.

Відсутність ортогональності концепцій. Надання можливості користувачу визначення власних понять мови моделювання також дозволяє побудувати мінімальну систему понять, що є семантично ортогональними. Але зазначимо, що ця задача повністю лежить на плечах користувача.

Неточна семантика. Саме тому дуже важливим є формальне визначення як моделей ПрО, так і призначених для їх породження метамodelей (мов моделювання). Такий підхід розкритий у наших інших роботах, де метамodelі будуються на основі певних математичних теорій (теорії графів, векторної алгебри, логіки та ін.).

Обмеження користувача. Залежність від об'єктно-орієнтованої парадигми. Складність стикування з мовами розробки. Важливим аспектом нашого підходу є те, що предметно-орієнтована модель не містить у собі архітектурних рішень, а також не управляються визначеними заздалегідь графічними нотаціями чи парадигмами програмування. Користувач сам буде графічну нотацію та обирає методологію (процес) моделювання.

Складність розширень. У цьому випадку модель ПрО обмежена лише метамodelлю, з якої вона породжена. Саме тому метамodelь визначає широту класу задач, що можна розв'язати у межах даного підходу. Інша ідея пропонованого нами підходу полягає у наданні можливості побудови метамodelі у рамках формалізму ще більшого рівня абстракції (мета-метамodelі). Такий підхід дозволяє користувачу побудувати власну мову моделювання ПрО.

Проблеми при вивченні й впровадженні. Проблеми вивчення мови моделювання ПрО мінімізуються або повністю зникають, так ця мова розроблюється власне фахівцем ПрО.

Неможливість зміни процесу моделювання. Разом із наданням можливості визначення системи понять, в яких буде здійснюватися моделювання ПрО, ідея нашого підходу полягає у моделюванні процесу розробки системи. Таким чином, користувач може сам будувати модель процесу розробки системи, що відповідає конкретним вимогам організації чи проекту.

Ефективність коду. Користувач також може будувати модель ПрО, що має будь-які визначені

заздалегідь властивості, зокрема повноту за Тьюрингом. Цей підхід залежить від ефективності роботи так званих генераторів коду.

Висновки

У статті дана загальна характеристика уніфікованої мови моделювання UML, проаналізовані її переваги та вади, визначені конкретні шляхи подолання недоліків «універсального» підходу до моделювання ПрО.

Сутність підходу полягає у наданні можливості користувачу (фахівцю ПрО, зокрема у галузі систем озброєння та військової техніки) розробки власних мов моделювання. Такі предметно-орієнтовані мови моделювання будуються на основі визначених математичних формалізмів (теорії графів, теорії множин, векторної алгебри та ін).

Принципи побудови предметно-орієнтованих мов моделювання, призначених, зокрема, для проектування систем озброєння й військової техніки, будуть розглянуті у наших наступних роботах.

Список літератури

1. Russ Miles and Kim Hamilton. *Learning UML 2.0*. – O'Reilly Media. – 2006. – 288 p.
2. Tim Weikiens. *Systems Engineering with SysML/UML. Modeling, Analysis, Design*. – Morgan Kaufmann Publishers Inc. – 1st edition, 2008. – 320 p.
3. Фаулер М., Скотт К. *UML. Основы. Краткое руководство по унифицированному языку моделирования. Второе издание*. – М.: Символ-Плюс, 2002. – 192 с.
4. Крэг Ларман. *Применение UML 2.0 и шаблонов проектирования*. – М.: Вильямс, 2006. – 736 с.
5. Джозеф Шмюллер. *Освой самостоятельно UML 2 за 24 часа. Практическое руководство*. – М.: Вильямс, 2005. – 416 с.
6. Грейди Буч, Джеймс Рамбо, Айвар Джекобсон. *Язык UML. Руководство пользователя*. – М., СПб.: ДМК Пресс, Питер, 2004. – 432 с.

7. Буч Г., Якобсон А., Рамбо Дж. *UML. Классика CS. 2-е изд. / Пер. с англ.; Под общей редакцией проф. С. Орлова* – СПб.: Питер, 2006. – 736 с.

8. Роберт Франс, Судипто Гош, Транг Дин-Тронг, Эрнор Соулберг. *Разработка на базе моделей с использованием UML 2.0: обещания и просчеты // Открытые системы, №03*. – 2006.

9. B. Henderson-Sellers; C. Gonzalez-Perez. *Uses and Abuses of the Stereotype Mechanism in UML 1.x and 2.0 // Model Driven Engineering Languages and Systems. Springer Berlin / Heidelberg*.

10. *UML Forum*. [Електрон. ресурс]. – Режим доступу. - <http://www.uml-forum.com/>

11. *Rational Software*. [Електрон. ресурс]. – Режим доступу. - <http://www-01.ibm.com/software/rational/>

12. *Object Management Group*. [Електрон. ресурс]. – Режим доступу. - www.omg.org

13. *Model Driven Architecture*. [Електрон. ресурс]. – Режим доступу. - www.omg.org/mda

14. Разработка на базе моделей с использованием UML 2.0: обещания и просчеты // *Открытые системы. 2006. - № 3*. [Електрон. ресурс]. – Режим доступу. - <http://www.osp.ru/os/2006/03/1156601/>

15. А.В. Бабич. *Введение в UML*. [Електрон. ресурс]. – Режим доступу. - <http://www.intuit.ru/departments/intuml/>

16. *Systems Modeling Language*. [Електрон. ресурс]. – Режим доступу. - <http://www.omg.sysml.org>

17. *Каталог UML профилей*. [Електрон. ресурс]. – Режим доступу. - http://www.omg.org/technology/documents/profile_catalog.htm

18. Александр Ложечкин. *UML или DSL? Унификация или нацеленность на задачу*. [Електрон. ресурс]. – Режим доступу. - <http://www.gotdotnet.ru/blogs/allo/595/>

Надійшла до редколегії 9.06.2010

Рецензент: д-р фіз.-мат. наук, проф. В.В. Кідалов, Бердянський державний педагогічний університет, Бердянськ.

ЦЕЛЕСООБРАЗНОСТЬ ПРИМЕНЕНИЯ UML ДЛЯ МОДЕЛИРОВАНИЯ СИСТЕМ ВООРУЖЕНИЯ И ВОЕННОЙ ТЕХНИКИ

В.И. Межуев

Предметом рассмотрения статьи является унифицированный язык моделирования UML (Unified Modeling Language), в частности, целесообразность его применения для проектирования систем вооружения и военной техники. Дана общая характеристика UML, проанализированы его преимущества и изъяны, определены пути преодоления недостатков «универсального» подхода к моделированию предметных областей.

Ключевые слова: унифицированный язык моделирования UML; предметно-ориентированное моделирование.

EXPEDIENCY OF UML APPLICATION FOR DESIGNING SYSTEMS OF ARMAMENT AND MILITARY TECHNIQUE

V. I. Mezhuiev

The Unified Modelling Language (UML), especially expediency of its application for designing systems of armament and military technique is a subject of consideration in the paper. The general characteristic of UML, its advantages and flaws is given, the ways of overcoming deficiencies of the "universal" approach to modelling of subject domains are defined.

Keywords: Unified Modelling Language UML; domain-specific modelling.