

УДК 681.3.068

Гавран В. –ст. гр. КСМм – 51

Тернопільська Академія Народного Господарства

МЕТОД ТРАНСФОРМАЦІЇ МЕТАМОДЕЛІ UML

Науковий керівник: к. ф.-м. н., доцент Касянчук М. М.

Моделювання майбутнього програмного продукту використовується при об'єктно-орієнтованому підході у програмуванні, коли розв'язуються складні задачі та створюються масштабні програмні системи. Модель дозволяє наочно відобразити структуру і поведінку системи.

У наш час застосовується підхід до моделювання програмного забезпечення з використанням уніфікованої нотації, яка ґрунтується на використанні Уніфікованої Мови Моделювання (UML – Unified Modeling Language).

UML призначена для зображення діаграм, що ілюструють модель системи, а також дозволяє на основі графічної моделі системи сформувати об'єктну ієрархію класів.

Запропонований метод полягає в переході від логічної до фізичної метамоделі UML. Логічна метамодель описана в стандарті мови і є повною й точною.

Основним недоліком існуючих прикладних інструментальних засобів є те, що вони незалежно будують свою об'єктну модель, але не доводять відповідність моделі стандарту. Їх моделі є, як правило, неповними, хоч і пристосовані для рішення прикладних задач.

Тому для гарантії відповідності переходу від логічної до фізичної метамоделі UML потрібно трансформувати модель стандарту так, щоб на кожному кроці зберігалася відповідність і додавалися необхідні властивості. Даний підхід будемо називати трансформаційним.

Перетворення виконується за допомогою систематичного застосування зразків проектування (patterns). Якщо кожен з них зберігає семантику і додає потрібну властивість, то в результаті буде повна й точна модель із необхідними властивостями.

Зразок проектування описує задачу, яка виникає в процесі роботи, а також принцип її рішення, причому таким чином, що це рішення можна використати багато разів.

Для одержання нової трансформованої моделі можна використати такі зразки проектування:

- *Елімінація множинного наслідування (Avoiding Multiple Inheritance)*. Даним зразком проектування усуваються проблеми, пов'язані із множинним наслідуванням.
- *Грань (Facet)*. Унікається створення великого класу з інтерфейсом, у якого величезна кількість логічно невпорядкованих методів.
- *Компонувальник (Composite)*. Компонує об'єкти у деревоподібні структури для подання ієрархій.
- *Одинак (Singleton)*. Гарантує, що в класі є тільки один екземпляр, і надає до нього глобальну точку доступу.

Отже, в результаті розроблений трансформаційний метод переходу від логічної до фізичної метамоделі UML. Він дозволяє реалізовувати повну та точну об'єктну метамодель UML з необхідними властивостями на будь-якій об'єктно-орієнтованій мові програмування.