

СИСТЕМА МОДЕЛЕЙ ПРЕДМЕТНОЇ ОБЛАСТІ «МЕТАТЕХНОЛОГІЯ ПРОГРАМУВАННЯ»

Є.І. Моренцов

Розглянуто спектр моделей предметної області «Метатехнологія програмування», які використовуються при конструюванні сервісно-орієнтованих технологій програмування в семантичному веб-середовищі. Визначені базові терміни і поняття цієї предметної області. Наведено концептуальну модель технології програмування й концептуальну та онтологічну моделі метатехнології програмування.

Ключові слова: метатехнологія програмування, концептуальна модель, онтологічна модель, сервісно-орієнтована технологія програмування, семантичний веб.

Рассмотрен спектр моделей предметной области «Метатехнология программирования», которые используются при конструировании сервисно-ориентированных технологий программирования в семантической веб-среде. Определены базовые термины и понятия этой предметной области. Приведены концептуальная модель технологии программирования, концептуальная и онтологическая модели метатехнологии программирования.

Ключевые слова: метатехнология программирования, концептуальная модель, онтологическая модель, сервисно-ориентированная технология программирования, семантический веб.

The spectrum of models of domain «Software Metatechnology» which are used for the construction of the software service-oriented technologies in the semantic web environment is considered. The basic terms and concepts of this domain are defined. The conceptual model of software technology, conceptual and ontological models of a software metatechnology are given.

Key words: software metatechnology, conceptual model, the ontological model, software service-oriented technology, semantic web.

Вступ

Головним завданням метатехнології програмування як напрямку в програмній інженерії [1] є створення нових технологій програмування (ТП) для розробки програмного забезпечення (ПЗ), яке вирішує конкретні задачі автоматизації певної предметної області (ПрО). На цей час перспективними є сервісно-орієнтовані технології програмування (СОТП), які дозволяють вести розробку ПЗ у розподіленому середовищі і використовувати компоненти технології, представлені у якості семантичних веб-сервісів. Конструювання технології відбувається на етапах метатехнологічного процесу, який базується на концепції інтелектуалізованого створення СОТП у семантичному веб-середовищі [2]. В основі цієї концепції лежить *сервісно-орієнтована парадигма* у поєднанні з *онтологічно-базовий підходом* до створення технологій розробки ПЗ. При такому підході компоненти технології розміщені у веб-середовищі як веб-сервіси, а онтології використовуються для пошуку і добору компонентів. Використання онтологічно-базованого підходу при створенні СОТП вимагає розробки системи моделей ПрО «Метатехнологія програмування», зокрема концептуальної та онтологічної моделей як основних.

Базові терміни і поняття предметної області «Метатехнологія програмування»

В контексті вирішуваної задачі конструювання СОТП базовими поняттями предметної області «Метатехнологія» є наступні.

Метатехнологія – довільний процес впливу на ТП, що перетворює її від деякого початкового до деякого кінцевого стану. Можна говорити про метатехнологію створення технологій (креативна метатехнологія), метатехнологію розвитку технологій (upgrade-метатехнологія), і, нарешті, метатехнологію впровадження технологій (invade-метатехнологія).

Технологія програмування – це сукупність технологічних процесів, методів, нотацій і засобів, використовуваних у процесі розробки програмного забезпечення. Технологічний процес складається з технологічних операцій і являє собою набір технологічних інструкцій, що включають:

- вказування послідовності виконання технологічних операцій;
- перерахування умов, при яких виконується та або інша операція;
- опису самих операцій, де для кожної з них визначені вихідні дані, результати, а також інструкції, нормативи, стандарти, критерії й методи оцінки і т. п.

Крім набору операцій та їх послідовності, технологія також визначає спосіб фіксації проектних рішень у процесі розробки (опису розроблювальної системи) — нотації, використовувани на конкретному етапі розробки.

Будь-яка технологія програмування базується на деякій методології розробки програм. Тому кожна технологічна операція (або група операцій) базується на використанні певних методів і засобів.

Технологічний процес – сукупність взаємозалежних технологічних операцій.

Технологічна операція – основна одиниця роботи, виконувана визначеною роллю, яка:

а) має на увазі чітко визначену відповідальність ролі;

б) дає чітко визначений результат (набір артефактів розробки ПЗ), що базується на визначених вхідних даних (іншому наборі артефактів);

в) являє собою одиницю роботи з жорстко визначеними границями, які встановлюються при плануванні проекту.

Технологічний процес розробки програмного забезпечення (software process) – це визначена послідовність технологічних операцій, які визначають напрямки діяльності з розробки ПЗ.

Метод у програмній інженерії – це сукупність способів і прийомів виконання технологічних операцій у процесі розробки ПЗ.

Нотація – засіб фіксації прийнятих рішень, отриманих результатів і стану ПЗ у процесі його розробки.

Інструментальний засіб – програмний засіб, що забезпечує автоматизовану підтримку діяльності, яка виконується в рамках технологічних операцій.

Розроблювальне програмне забезпечення – інформаційна або матеріальна сутність, яка створюється, модифікується або використовується в деякій технологічній операції (модель, документ, код, тест і т. п.). В процесі розробки ПЗ послідовно змінює свої стани, переходячи з вхідного до вихідного, як в рамках однієї технологічної операції, так і в рамках всього процесу розробки.

Стан розроблювального ПЗ – сукупність артефактів, які характеризують, описують та ідентифікують ПЗ на деякому етапі процесу розробки.

Роль – визначення поведінки й обов'язків окремої особи або групи осіб у середовищі організації-розроблювача ПО, що здійснюють діяльність у рамках деякого технологічного процесу й відповідальних за певні робочі продукти.

Настанова – практичний посібник з виконання однієї або сукупності технологічних операцій. Настави включають методичні матеріали, інструкції, нормативи, стандарти й критерії оцінки якості робочих продуктів.

Класифікація методів та інструментів програмної інженерії, яка використана у нашому дослідженні, викладена у матеріалах по SWEBOOK [3, розд. 9].

Моделі домену метатехнології

Система моделей домену «Метатехнологія програмування» містить:

- спектр концептуальних моделей, до якого входять модель технології програмування, що задає основні компоненти (поняття) технології і зв'язки між ними, та модель сумісності компонентів технології програмування, яка використовується при об'єднанні компонентів у єдину СОПІ;

- розширену семантичну модель інструмента як веб-сервісу, оскільки класичної четвірки множин, що задає семантичну модель, у випадку з інструментом як веб-сервісом недостатньо;

- концептуальну модель предметної області «Метатехнологія програмування», яка є визначальною серед усього спектру концептуальних моделей домену метатехнології і представлена у вигляді семантичної мережі;

- онтологічну модель предметної області «Метатехнологія програмування». Ця модель є визначальною і представлена у вигляді UML-діаграми класів.

У цій роботі ми розглянемо концептуальні моделі технології програмування і ПрО «Метатехнологія програмування», а також онтологічну модель даної предметної області.

Концептуальна модель технологій програмування

У класичному понятті **концептуальна модель** розуміється як модель представлена множиною понять і зв'язків між ними, які визначають сенс предметної області, що розглядається, або її конкретного об'єкта.

Концептуальна модель технології програмування показана на рис. 1.

Основними її компонентами – поняттями даної моделі – є технологічний процес, операції технологічного процесу, методи, нотації, інструментальні засоби та стани розроблюваного ПЗ.

Технологічний процес – задає послідовність дій по створенню програмного засобу і містить **Операції технологічного процесу**, кожна з яких є чітко визначеною дією, яка **перетворює** розроблюване ПЗ з одного стану в інший. Операція технологічного процесу **виконується** **Методом** і **автоматизується** **Інструментальним засобом**. Таким чином при виконанні технологічного процесу ПЗ послідовно

перетворюється у послідовність станів – від постановки задачі до готового програмного продукту. *Стан розроблюваного ПЗ* фіксується за допомогою *Нотацій*.

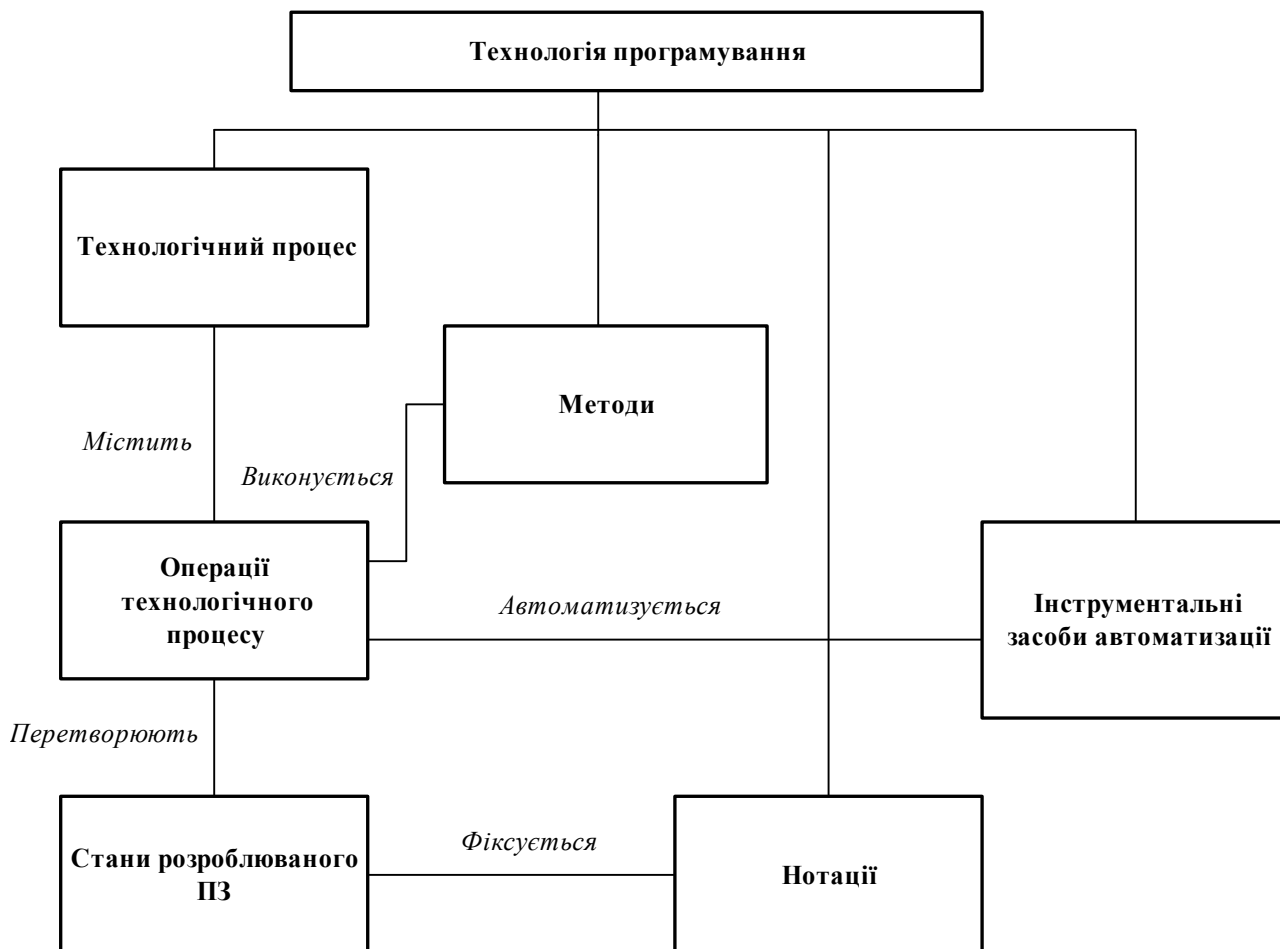


Рис. 1. Концептуальна модель технології програмування

Концептуальна модель предметної області «Метатехнологія програмування»

Концептуальна модель ПрО «Метатехнологія» в контексті вирішуваної задачі показана на рис. 2 у вигляді семантичної мережі.

На ній представлені наступні класи об'єктів:

- 1) сервісно-орієнтована технологія програмування, призначена для розробки програмних засобів;
- 2) розробники програмного засобу, які використовують СОТП для розробки ПЗ;
- 3) технологічні процеси;
- 4) методи розробки ПЗ;
- 5) нотації, якими фіксуються стани ПЗ в процесі розробки;
- 6) інструментальні засоби, які підтримують методи і нотації;
- 7) операції технологічного процесу;
- 8) стани розроблюваного ПЗ.

Оскільки назви класів співпадають з базовими поняттями ПрО «Метатехнологія», які були описані вище, дамо лише семантику зв'язків між класами.

Зв'язок «*Використовує для*» між класами Розробники і СОТП означає, що будь який розробник ПЗ використовує СОТП в процесі розробки ПЗ, застосовуючи всі компоненти СОТП. Той же самий зв'язок між класами Методи і Нотації означає, що метод використовує нотацію для фіксації результатів свого виконання, тобто деякого стану ПЗ

Семантика зв'язку «*Складається із*» між класом СОТП з одного боку і класами Технологічні процеси, Методи, Нотації, Інструментальні засоби з іншого боку полягає в тому, що будь-який екземпляр

класу СОТП містить в собі екземпляри інших перелічених класів як компоненти. Тобто маємо зв'язок типу композиція.

Зв'язок «*Містить*» між класами Технологічні процеси і Технологічні операції означає, що екземпляр першого класу складається із певного числа екземплярів другого.

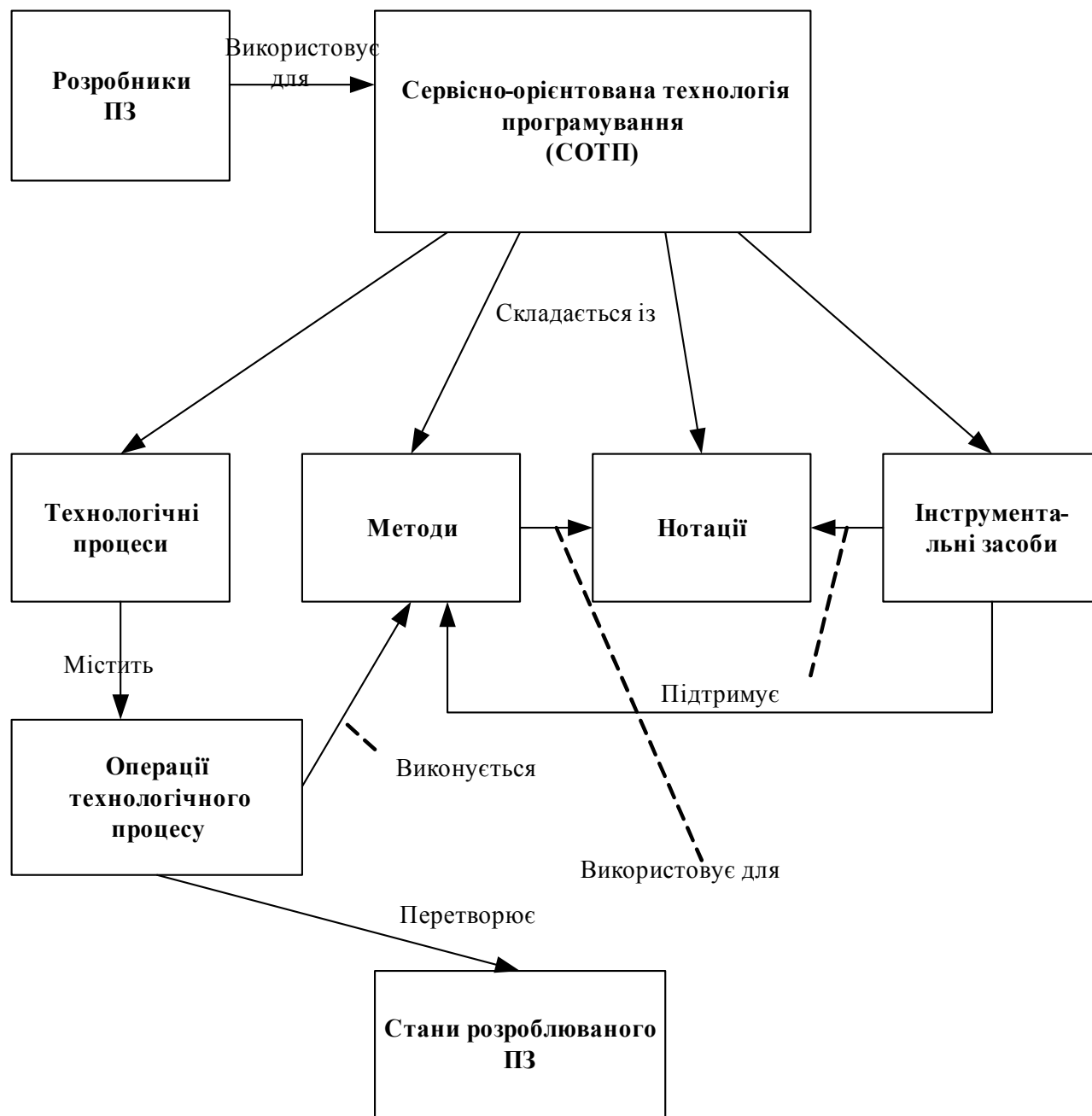


Рис. 2. Концептуальна модель ПрО «Метатехнологія програмування»

Семантика зв'язку «*Виконується*» між класами Технологічні операції і Методи полягає в тому, що будь-яка технологічна операція виконується відповідним методом.

Зв'язок «*Перетворює*» між класами Технологічні операції і Стани розроблюваного ПЗ означає, що виконання технологічної операції переводить розроблюване ПЗ із деякого проміжного стану в інший проміжний стан в процесі розробки. При цьому кінцевий стан ПЗ фактично є готовим ПЗ.

Зв'язок «*Підтримує*» між класами Інструментальні засоби і Методи, а також Інструментальні засоби і нотації означає, що інструментальний засіб автоматизує виконання методу і фіксацію результатів у відповідному операційному середовищі.

Онтологічна модель предметної області «Метатехнологія програмування»

Концепція розробки онтології метатехнології програмування. Фактично першою фундаментальною роботою в області зводу знань по програмній інженерії був проект SWEBOK [3]. У ньому не дійшло до представлення онтологій даної предметної області, але він представив добре структуровані і найбільш повні на той момент знання з програмної інженерії. Проект закінчив своє існування у 2006 році, коли розпалася група, що його виконувала. На думку автора однією з причин цього був занадто жорсткий підхід до структуризації знань у програмній інженерії, що по мірі розвитку цієї області не давало змоги адекватно представити нові знання, які не завжди можна було вкласти у запропоновану структуру. Також виглядає дивним той факт, що, незважаючи на достатньо повний звід знань з програмної інженерії, представлений у SWEBOK, так і не було розроблено онтологічних моделей представлення цих знань. Виходом з цієї ситуації може бути перереструктуризація знань шляхом об'єднання концепції представлення знань у SWEBOK із представленням знань про ТП у метатехнології програмування, як складової частини дисципліни «Програмна інженерія», і застосування онтологій для представлення знань у цій ПрО. Саме такий підхід був запропонований і взятий за основу в Інституті програмних систем НАН України.

У цій статті будемо розуміти під *онтологією* формальний явний опис *понять* у розглянутій предметній області «Метатехнологія програмування» (тобто *класів*), властивостей кожного поняття, що описують різні властивості й атрибути поняття (*слотів*, іноді їх називають *ролями* або *властивостями*), і обмежень, накладених на слоти (*фацетів*, іноді їх називають *обмеженнями ролей*). Онтологія разом з набором індивідуальних *екземплярів класів* утворює *базу знань*.

У центрі онтології ПрО «Метатехнологія програмування» перебувають класи. Класи описують поняття предметної області. Наприклад, клас методів представляє всі методи розробки програмного забезпечення. Конкретні методи – екземпляри цього класу. Клас може мати підкласи, які представляють більш конкретні поняття, ніж надклас. Наприклад, ми можемо розділити клас методів на методи аналізу, проектування, тестування. Слоти описують властивості класів і екземплярів: метод проектування Х – призначений для проектування інформаційних систем і його автором є N. Тут у нас є два слоти, які описують метод у цьому прикладі: слот «призначення» зі значенням «проектування інформаційних систем» і слот «автор» (розроблювач) зі значенням «N». Також на рівні класу в екземплярів класу «Метод» є слоти, які описують усі його інші властивості.

На практиці розробка онтології ПрО «Метатехнологія» включає:

- 1) визначення класів в онтології;
- 2) розташування класів у таксономічну ієрархію (підклас – надклас);
- 3) визначення слотів і опис, що допустимих значень цих слотів;
- 4) заповнення значень слотів екземплярів.

Після цього можна створити базу знань, визначивши окремі екземпляри цих класів, ввівши в певний слот значення й додаткові обмеження для слота.

Для побудови онтологічної моделі ПрО «Метатехнологія програмування» був використаний *структурний (або когнітивний)* підхід, у якому побудова моделі здійснюється шляхом виділення елементів предметної області, їх взаємозв'язків і семантичних відносин. Виділені поняття ПрО утворюють систему, під якою розуміється сукупність понять, що володіє наступними властивостями:

- унікальністю (відсутністю надмірності);
- повнотою (досить повним описом різних процесів, фактів, явищ і т. д. предметної області);
- вірогідністю (валідністю – відповідністю виділених одиниць значеннєвої інформації їх реальним найменуванням) і несуперечністю (відсутністю омонімії).

Структурний підхід до побудови онтологічної моделі добре поєднується із концепцією розробки онтології ПрО «Метатехнологія програмування», яка ґрунтується на поєднанні структури СОТП із знаннями у SWEBOK. У контексті вирішуваної задачі виділення елементів ПрО області, їх взаємозв'язків і семантичних відносин проведено, виходячи із CASE-подібної моделі СОТП, яка визначає також її структуру.

Технологія програмування (розробки ПЗ) в узагальненому вигляді є набором операцій, методів, нотацій та інструментів, які застосовуються групою розробників для створення і підтримки ПЗ. У СОТП інструментальні засоби представляються як веб-сервіси, що знаходяться у семантичному веб-середовищі. При створенні СОТП інтегруються всі її компоненти. Істотну частину знань про компоненти СОТП можна взяти із SWEBOK, особливо в частині методів та інструментів, хоча й вони далеко не повні для потреб нашого дослідження. До того ж структуризація знань проводиться у рамках побудованої нами онтологічної моделі.

Границі побудованої онтології визначаються ПрО «Метатехнологія», вирішуваною задачею конструювання СОТП і моделлю представлення СОТП. При цьому поняття, які присутні в ПрО, але не використовуються при вирішенні задачі конструювання СОТП, не будуть присутні у онтологічній моделі.

При створенні початкової онтології метатехнології програмування в ПрО «Програмна інженерія» був виділений домен «Метатехнологія програмування» і сформована його онтологія в контексті задачі добору компонентів технології програмування. При цьому підхід полягав у аналізі використовуваних для розробки ПЗ онтологій і виборі з них таких, які можна використати на рівні метатехнології, а також у розробці нових онтологій, які необхідні для вирішення задачі конструювання СОТП.

Виходячи з вище визначених понять і класів, у ПрО «Метатехнологія» виділимо класи, які будуть необхідні нам в процесі конструювання СОТП:

- 1) клас СОТП;
- 2) клас Технологічні процеси;
- 3) клас Методи;
- 4) клас Нотації;
- 5) клас Інструментальні засоби;
- 6) клас Технологічні операції.

У порівнянні з концептуальною моделлю у нас випало два класи – «Розробники ПЗ» і «Стани розроблюваного ПЗ», які фактично не грають ніякої ролі в конструюванні СОТП. Для кожного класу визначені слоти (властивості), які показані на рис. 3, і фацети (обмеження, які накладені на значення слотів).

Перший клас – це *клас СОТП*. Слот «*Назва*» містить назву СОТП як екземпляра класу, слот «*Розробник*» – дані про розробника СОТП, слот «*Призначення*» – для розробки яких видів ПЗ призначена СОТП, слоти «*Використовуваний технологічний процес*», «*Використовувані методи*», «*Використовувані нотації*», «*Використовувані інструментальні засоби*» містять назви відповідно ТП, методів, нотацій та інструментальних засобів, що є компонентами даної СОТП, крім того ці слоти використовуються для встановлення зв'язку із СОТП та її компонентами.

Другий клас – *Технологічні процеси* підрозділяється на два підкласи:

- 1) процеси технічної діяльності;
- 2) процеси управлінської діяльності.

Перший з них містить процеси, пов'язані з виконання технологічних операцій, які переводять ПЗ із одного стану в інший, а другий – процеси з управління розробкою ПЗ.

Клас «Технологічні процеси» містить наступні слоти. Слот «*Назва*» ідентифікує ТП, слот «*Призначення*» описує для виконання яких крупних робіт призначений процес, слот «*Технологічні операції*» задає перелік ТО, які входять до цього процесу і встановлює зв'язок між класами «Технологічні процеси» і «Технологічні операції». Підклас «Процеси технічної діяльності» має додатковий слот «*Етап (фаза) технологічного процесу*», де зазначено на якому етапі ЖЦ застосовується процес.

Третій клас – *Технологічні операції* має наступні слоти. Слот «*Назва*» містить назву ТО в контексті дії, яка виконується на цій операції. Слоти «*Вхідний стан ПЗ*» і «*Вихідний стан ПЗ*» описують відповідно вхідні і вихідні дані операції. Слот «*Використовуваний метод*» визначає метод, яким виконується операція, а також встановлює зв'язок між класами «Технологічні операції» і «Методи».

Слоти четвертого класу – *Методи* задають наступне. Слот «*Назва*» – назва метода, яка однозначно його ідентифікує, слот «*Призначення*» – для виконання якої роботи з розробки ПЗ призначений метод, слот «*Використовувана нотація*», яку саме нотацію використовує метод для фіксації результатів своєї роботи, а також встановлює зв'язок між класами «Методи» і «Нотації». Слоти «*Вхідний стан ПЗ*» і «*Вихідний стан ПЗ*» задають вхідні і вихідні дані методу відповідно.

У п'ятому класі – *Нотації* слот «*Вид*» задає вид нотації (мовна, графічна), слот «*Назва*» ідентифікує нотацію, слот «*Призначення*» описує можливе застосування цієї нотації у процесі розробки ПЗ.

Найбільше слотів має клас *Інструментальні засоби*. Слот «*Назва*» ідентифікує інструментальний засіб, слот «*Призначення*» описує для виконання яких робіт з розробки ПЗ може використовуватися цей інструмент, слот «*Виробник*» містить дані про розробника цього ІЗ, слот «*Місцезнаходження у мережі*» вказує де саме (за URL) можна знайти даний інструмент. Два слоти «*Метод, що підтримується*» і «*Нотація, що підтримується*» вказують, відповідно, виконання якого саме методу автоматизує інструментальний засіб і яку саме нотацію він використовує. Ці два слоти також служать для встановлення зв'язку між класом «Інструментальний засіб» з одного боку і класами «Методи» і «Нотації» з іншого. Цей клас може також мати

додаткові слоти, які відповідають критеріям якості встановленим стандартами «ISO/IEC 14102:2008. Information Guideline for the evaluation and selection of CASE tools» та «ДСТУ 3919-1999. Інформаційні технології. Основні напрямки оцінки та відбору CASE-інструментів» [4].

СОПІ
Назва
Розробник
Призначення (вид розроблюваного ПЗ)
Використовувані методи
Використовувані нотації
Використовувані інструментальні засоби

Технологічні процеси
Назва
Призначення
Технологічні операції

Процеси технічної діяльності
Етап (фаза) технологічного процесу

Процеси управлінської діяльності

Технологічні операції
Назва
Вхідний стан ПЗ
Вихідний стан ПЗ
Використовуваний метод

Методи
Назва
Призначення
Використовувана нотація
Технологічна операція
Вхідний стан ПЗ
Вихідний стан ПЗ

Інструментальні засоби
Назва
Призначення
Виробник
Місцезнаходження у мережі (URL)
Технологічна операція
Метод, що підтримується
Нотація, що підтримується
Вхідний стан ПЗ
Вихідний стан ПЗ
Додаткові слоти за ISO/IEC 14102 – ДСТУ 3319-1999

Нотації
Вид
Назва
Призначення

Рис. 3. Класи Про «Метатехнологія програмування» та їхні слоти

Онтологічна модель у вигляді UML-діаграми класів представлена на рис. 4. У ній для компактності і наглядності у класах представлені тільки слоти, через які здійснюється зв'язок між класами. Представлену модель ми визначили як початкову, оскільки вона у подальшому може змінюватися і доповнюватися.

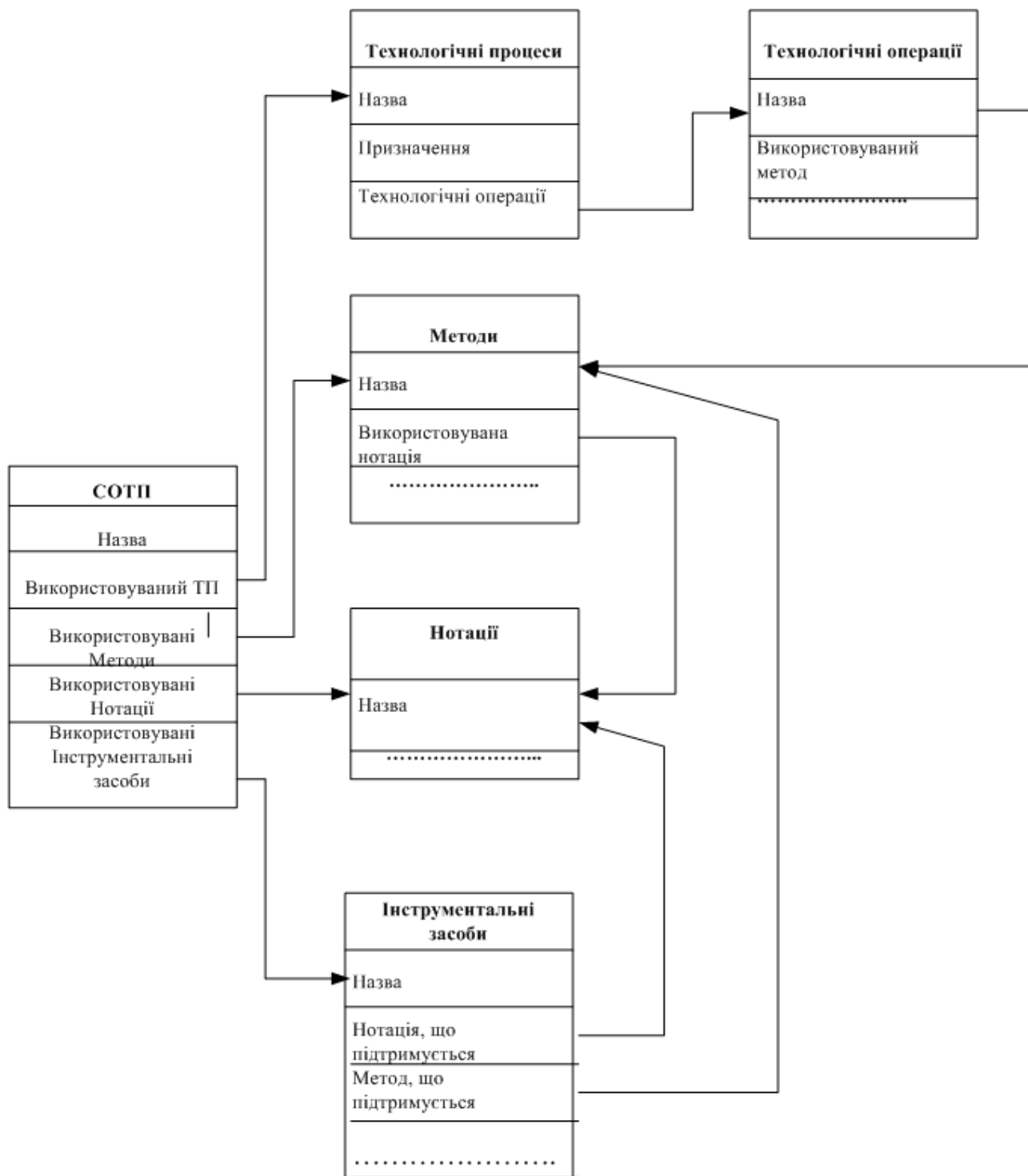


Рис. 4. Онтологічна модель ПрО «Метатехнологія»

Висновки

Визначена і розроблена система моделей ПрО «Метатехнологія програмування», яка використовується при конструюванні якісно нових сервісно-орієнтованих технологій програмування. В рамках цієї системи у контексті основної задачі метатехнології розроблена онтологічна модель ПрО «Метатехнологія», яка є

визначальною в системі моделей і в подальшому може використовуватись для побудови відповідної онтології шляхом інстанціювання, тобто створення екземплярів класів.

Література

1. Моренцов Е.И. Метатехнология и концепции создания специализированных технологий программирования. *Проблемы программирования*. 1998. Вып. 3. С. 15–22.
2. Моренцов С.І. Концепція інтелектуалізованого створення сервісно-орієнтованих технологій програмування в семантичному веб-середовищі. *Проблеми програмування*. 2017. № 2. С. 106–116.
3. Alain Abran, James W. Moore. Guide to the Software Engineering Body of Knowledge (2004 edition, PDF version only). [Електронний ресурс]. Режим доступу: <http://www.computer.org/portal/web/swebok>
4. ДСТУ 3919-1999. Інформаційні технології. Основні напрямки оцінювання та відбору CASE-інструментів. Київ: Держстандарт України, 2000. 39 с.

References

1. Morentsov Y. I. (1998) Metatechnology and concepts of creation of specialized software technologies. *Problems in programming*. (3). P. 15–22. (in Russian).
2. Morentsov Y. I. (2017) The concept of intellectualized creation of service-oriented technologies of software development in the semantic web environment. *Problems in programming*. (2). P. 106–116. (in Ukrainian).
3. Alain Abran, James W. Moore. Guide to the Software Engineering Body of Knowledge (2004 edition, PDF version only). — [Electronic resource]. Mode of access: <http://www.computer.org/portal/web/swebok>.
4. DSTU 3919:1999. (2000) Information technology. Guideline for the evaluation and selection of CASE tools. Kyiv: Derzhstandart Press. (in Ukrainian).

Про автора:

Моренцов Євгеній Іванович,

кандидат технічних наук, старший науковий співробітник.

Кількість наукових публікацій в українських виданнях – 45.

Кількість наукових публікацій в зарубіжних виданнях – 2.

Індекс Хірша – 1.

<http://orcid.org/0000-0002-0497-11838>.

Про місце роботи автора:

Інститут програмних систем НАН України,
03187, Київ, проспект Академіка Глушкова, 40.

Тел.: 526 42 86.

E-mail: yevhen18@ukr.net