

## КОНЦЕПЦІЯ ІНТЕЛЕКТУАЛІЗОВАНОГО СТВОРЕННЯ СЕРВІСНО-ОРІЄНТОВАНИХ ТЕХНОЛОГІЙ ПРОГРАМУВАННЯ В СЕМАНТИЧНОМУ ВЕБ-СЕРЕДОВИЩІ

Представлений підхід до створення принципово нових сервісно-орієнтованих технологій програмування. Розглянуто застосування онтологій при вирішенні цієї задачі і визначені класи онтологічних моделей для застосування у метатехнології програмування. Визначені основні етапи метапроцесу конструювання сервісно-орієнтованих технологій програмування та їхній зміст. Запропонована архітектура розподіленого середовища розробки програмного забезпечення, яка базується на хмарних технологіях. Ключові слова: метатехнологія програмування, сервісно-орієнтована технологія програмування, онтологія, семантичне веб-середовище.

### Вступ

**Метатехнологія програмування** – це комп'ютеризована технологія вищого рівня, призначена для побудови технологій програмування, які застосовуються для розробки програмного забезпечення (ПЗ) відповідного призначення. Головна задача метатехнології програмування – це створення конкретних технологій програмування, які базуються на існуючому наборі моделей життєвого циклу (моделей процесів розробки ПЗ), методів, нотацій та інструментальних засобів як метатехнологічних об'єктів. Технологія створюється шляхом добору потрібних об'єктів, які поєднуються в єдину технологічну систему, призначену для розробки визначених типів ПЗ.

В Україні перші фундаментальні дослідження з проблематики метатехнології як напрямку у програмній інженерії проведені в Інституті програмних систем НАН України кінець 1990-х – початок 2000-х років. Вони стосувалися створення MetaCASE-систем, що реалізують метатехнологію створення проблемно-орієнтованих технологій програмування (ПОТП) та інструментальних засобів її підтримки. Основним результатом цих досліджень стала методологія створення проблемно-орієнтованих технологій програмування методом конструювання з наявних технологічних об'єктів – компонентів ПОТП, представлених у базах знань [1]. Такий напрямок досліджень зумовлювався

тим, що на той час як основні перспективи розвитку CASE розглядалися об'єднання засобів баз знань та штучного інтелекту.

Однак, за останнє десятиріччя, з'явилися вагомні результати досліджень у галузі онтологій, які з успіхом почали застосовуватися у процесах розробки програмного забезпечення. Одночасно почали розвиватися веб-сервіси та «хмарні» технології у галузі комп'ютерних обчислень. Це дало поштовх дослідженням можливостей їх застосування у метатехнології для інтелектуалізованого створення якісно нових, сервісно-орієнтованих технологій програмування (СОТП). В статті викладена концепція інтелектуалізованого створення СОТП, в основі якої лежить сервісно-орієнтована парадигма у поєднанні з онтологічно-базованим підходом. Ці результати отримані при виконанні наукової роботи «Розробка теоретичних основ та прикладних питань побудови сервіс-орієнтованих прикладних програмних систем у семантичному веб-середовищі» в Інституті програмних систем НАН України.

### Застосування онтологій у програмній інженерії

У рамках програмної інженерії можна виділити дві основні групи онтологій. Перша – це онтології, які застосовуються в процесі розробки програмного забезпечення, друга – це онтології домену

метатехнології програмування, які мають використовуватись при створенні нових технологій програмування. І якщо перша група онтологій вже знайшла своє застосування, то використання другої нині знаходиться у стадії дослідження і розробки.

Основна мета онтології розробки ПЗ – забезпечити взаємодію між її розроблювачами, щоб оперувати знаннями розробки стандартного ПЗ, однозначно розуміти їх і виконувати визначені типи обчислень. Онтологія розробки ПЗ складається з екземплярів, що представляють визначені дані проекту; властивостей, які представляють бінарні відносини, збережені серед понять/екземплярів розробки програмного забезпечення; класів, що представляють поняття розробки програмного забезпечення, інтерпретовані як набори, що містять визначені дані проекту. Класи в онтології розробки програмного забезпечення створюються з описів понять розробки програмного забезпечення, що визначають умови, які мають бути задоволені даними проекту, щоб бути елементом класів.

Для того, щоб змодельовати домен розробки програмного забезпечення основні принципи онтології OWL включають онтологічні класи і підкласи, властивості (властивість типу даних і властивість об'єкта), асоціації між класами і властивостями, характеристиками властивостей, обмеженнями й екземплярами. Деталі цих компонентів онтологій OWL представлені в [2, 3].

Серед онтологій, використовуваних при розробці ПЗ, найбільш популярними є онтологія предметної області й онтологія задач. Керована онтологією розробка програмного забезпечення визначена як підхід, який заснований на онтологіях і бере до уваги семантичні обмеження, що адаптуються динамічним способом до нових обмежень. Онтології також використовуються як деякі артефакти в процесі розробки ПЗ.

**Онтологічно-базований підхід і сервісно-орієнтована парадигма у метатехнології програмування.** Основою концепції інтелектуалізованого створення

сервісно-орієнтованих технологій програмування (СОП) є онтологічно-базований підхід до створення технологій розробки ПЗ у поєднанні з сервісно-орієнтованою парадигмою.

**Онтологічно-базований підхід** до вирішення головної задачі метатехнології полягає у наступному. Для широко представлених у веб-середовищі множин методів, нотацій і інструментальних засобів розробки ПЗ створюються їх онтологічні (семантичні) описи. Також за допомогою онтологій специфікуються вимоги до створеної технології та її компонентів. Потім на основі цих описів і специфікацій виконується пошук потрібних компонентів, з яких формується потрібна технологія розробки ПЗ.

Важливо зазначити, що ми розглядаємо використання онтологій для конструювання технологій розробки ПЗ, а не для розробки власне його, тобто тільки у домені метатехнології.

Застосування **сервісно-орієнтованої парадигми** при створенні конкретної технології полягає у привнесенні у процес розробки як технології, так і ПЗ так званих «хмарних» технологій, коли компоненти технології, як правило – інструментальні засоби, являють собою веб-сервіси, розміщені у веб-середовищі. При такому підході розробнику ПЗ не треба витрачатися на придбання відповідної технології для кожної нової розробки або застосовувати одну й ту ж технологію, яка є в нього, для багатьох розробок. Він може використовувати потрібні йому компоненти технології як веб-сервіси за помірну ціну. Розробник може гнучко реагувати на зміну потреб в інструментах і технологіях, він самостійно визначає і змінює технологічні потреби.

**Технологія розробки ПЗ – це головний об'єкт метатехнології.** Використання сукупності її моделей, побудованих в рамках застосування онтологічно-базованого підходу і сервісно-орієнтованої парадигми, у процесі створення технології надає цьому процесу **інтелектуалізований характер.**

В узагальненому вигляді технологія програмування, яка використовується для розробки програмного забезпечення, представляється у вигляді четвірки

$$T_{\text{пр}} = \{P, M, N, T\},$$

де  $P$  – процес розробки ПЗ,  $M$  – множина методів розробки ПЗ,  $N$  – множина видів нотацій для представлення специфікацій ПЗ у процесі розробки,  $T$  – множина інструментальних засобів автоматизації процесу розробки ПЗ, які підтримують методи та нотації.

Конкретна технологія формується в процесі технологічної підготовки розробки шляхом створення технологічного процесу розробки і добору відповідних складових із множин  $M$ ,  $N$  і  $T$  на основі вимог до створюваної технології, які випливають із задач автоматизації конкретної предметної області (ПрО). І якщо раніше процес формування технології виконувався фактично вручну на основі аналізу та порівняння, то застосування онтологічно-базованого підходу і сервісно-орієнтованої парадигми наряду із семантик-веб дозволяють змістити його в бік інтелектуалізації.

Нині у веб-середовищі широко представлені множини методів, нотацій і інструментальних засобів розробки ПЗ. При цьому останні можна представити у вигляді веб-сервісів. Використання такого веб-сервісу і для розробника ПЗ, і для його власника буде більш вигідним, ніж купівля/продаж інструменту.

Однією з важливих задач при створенні технології програмування є добір її конкретних компонентів – методів, нотацій та інструментів. Якщо раніше задача добору вирішувалася вручну (хоча й були спроби використати бази знань), то застосування онтологій, спеціально розроблених для домену метатехнології програмування, надає додаткові переваги і слугує гарантією створення технології, яка найбільше задовольняє вимогам до неї.

В цьому контексті, однією з найважливіших задач є створення початкової онтології метатехнології

програмування. Тобто, беручи предметну область (ПрО) програмної інженерії, ми виділяємо в ній ПрО «Метатехнологія програмування» і формуємо її онтологію у контексті задачі добору компонентів технології програмування. При цьому наш підхід полягає у аналізі використовуваних для розробки ПЗ онтологій, виборі з них тих, які можна підняти на рівень метатехнології для застосування при створенні технологій програмування, а також у розробці нових додаткових онтологій, які необхідні для вирішення задачі створення СОТП у метатехнології.

Така СОТП створюється із компонентів які доступні у веб-середовищі. Ця технологія дозволить вести розподілену розробку програмних проектів в умовах, коли розробники географічно знаходяться у різних точках, взаємодіють між собою через мережу Інтернет, а інструментарій розробки ПЗ розташований у мережі як веб-сервіси. Оскільки при цьому технологія розробки ПС створюється з компонентів, які є веб-сервісами, то сама технологія становиться **сервісно-орієнтованою**, а застосування онтологій для пошуку і добору компонентів робить підхід до її створення **онтологічно базованим**. Для зберігання компонентів СОТП як сервісів та артефактів процесу розробки ПЗ пропонується застосування **хмарних технологій**.

### **Класи моделей у домені «Метатехнологія програмування»**

При інтелектуалізованому створенні технологій розробки ПЗ застосовуються чотири класи моделей.

**Перший клас** містить онтологічні моделі компонентів технології. За їхньою допомогою описуються, по-перше, вже існуючі компоненти технології, які розміщуються у веб-середовищі і є доступними як веб-сервіси, по-друге, вимоги до компонентів технології, що створюється. При цьому важливим є використання єдиної мови і єдиної системи понять, щоб у процесі пошуку потрібних компонентів пара-

метри пошуку могли бути співставлені з параметрами наявних компонентів.

**Другий клас моделей** – це онтологічні моделі, які описують сумісність компонентів технології як по горизонталі, так і по вертикалі. Якщо розглядати множини компонентів технології як відповідні класи, то горизонтальна сумісність є сумісністю екземплярів різних класів (сумісність у рамках однієї технологічної операції), а вертикальна сумісність – сумісністю екземплярів одного класу (сумісність на парі різних технологічних операцій). Саме сумісність як екземплярів різних класів, так і екземплярів одного класу дозволяє інтегрувати ці екземпляри у створювану технологію. Онтологічна модель сумісності компонентів технології може бути присутньою як у веб-середовищі, коли всі компоненти нової технології, що знаходяться у веб-середовищі добираються заново, так і у конструктора технології, коли вибір окремих компонентів зроблений апріорі. Тоді конструктор має задати модель сумісності для вже вибраних компонентів. Пошук і добір компонентів технології буде здійснюватися на основі онтологічних моделей компонентів і моделей сумісності компонентів.

**Третій клас моделей** – це модель платформи. До складу платформи входять інструментальні засоби – веб-сервіси, що включені до створеної технології, і використовуються для розробки ПЗ. Платформа дозволяє розробнику використовувати «хмарну» інфраструктуру в плані користування веб-сервісами, що надаються «хмарним» провайдером, а також для розміщення на ній артефактів у процесі розробки ПЗ. Відповідно модель платформи містить щонайменше онтологічні моделі інструментів платформи, включених до розробленої технології і артефактів розробки.

**Четвертий клас моделей** – це модель середовища розробки ПЗ. При

використанні «хмарних» технологій і сервісно-орієнтованої парадигми технологія розробки ПЗ стає інтегрованою і розподіленою у веб-середовищі. Що і впливає на визначення середовища розробки ПЗ, як системи технологічних, методичних та інструментальних засобів (компонентів технології), що використовуються для розробки ПЗ, і розміщених у веб-середовищі як веб-сервіси. Якщо всі компоненти розробленої технології знаходяться на одній платформі й на цій же платформі зберігаються артефакти розробки ПЗ, то до моделі середовища входить одна модель платформи. Якщо ж ці компоненти розміщені на декількох платформах, то до моделі середовища входять моделі всіх цих платформ. Модель середовища є інтегрованою моделлю і найвищою в ієрархії розглянутих моделей, оскільки містить моделі платформ, онтологічні моделі компонентів технології та моделі сумісності компонентів технології. Крім того до неї входить ще й **модель технологічного процесу розробки ПЗ**, яка визначає технологічні операції та методи, нотації й інструментальні засоби, що використовуються на цих операціях та взаємозв'язок між ними.

За допомогою моделі середовища, застосовуючи агенти ТП, можна в значному ступені автоматизувати керування процесом розробки ПЗ в частині визначення послідовності дій і застосування інструментів як веб-сервісів, які знаходяться у розподіленому середовищі.

**Концептуальна модель предметної області «Метатехнологія».** Серед усього спектру розроблених нами концептуальних моделей домену метатехнології програмування визначальною є концептуальна модель предметної області «Метатехнологія програмування». Вона містить множину понять і зв'язків між ними, які визначають сенс предметної області та/або її конкретний об'єкт, і представлена у вигляді семантичної мережі на рис. 1.

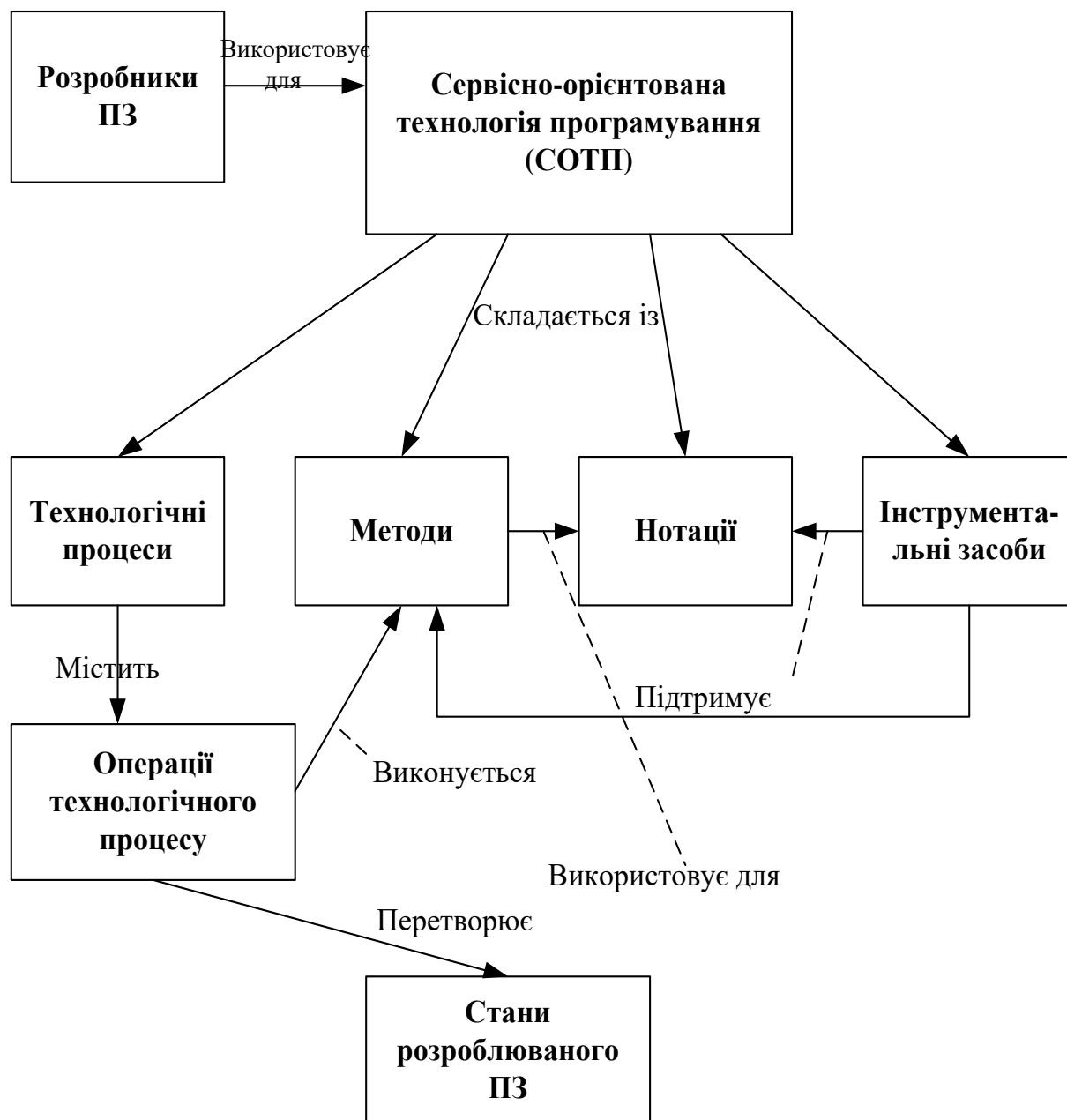


Рис. 1. Концептуальна модель ПрО «Метатехнологія»

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;

б) 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 у процесі розробки ПЗ, застосовую-

чи 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 у тому, що будь-який екземпляр класу СОТП містить у 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 класу.

Семантика зв'язку «Виконується» між класами Технологічні операції і Методи, та полягає у 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 у процесі розробки. При цьому кінцевий стан ПЗ фактично є готовим.

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 на комп'ютері у відповідному операційному середовищі.

### **Основні етапи і зміст метапроцесу конструювання СОТП**

Основою конструювання СОТП є процесний підхід, що включає у себе формування схеми ліній із процесів і операцій, необхідних для розробки компонентів майбутнього ПЗ. Ця схема являє собою технологічний процес розробки ПЗ і включає всі необхідні дії, які треба виконати від постановки завдання до отримання готового програмного засобу. При

цьому введемо одне обмеження: у процес розробки включаються процеси, що виконують дії, зв'язані тільки з розробкою ПЗ. Тобто процеси інсталяції й супроводу ПЗ ми не розглядаємо.

Головна вимога до побудови процесу розробки ПЗ – комплектування його із процесів ЖЦ, що відповідають розроблюваному домену й стосуються власне розробки ПЗ. У такий спосіб процес розробки ПЗ виходить фактично із ЖЦ ПЗ.

Потім цей процес деталізується на технологічні операції (ТО) і наповнюється методами, нотаціями та інструментами, які застосовуються розробниками для створення ПЗ.

Виходячи із представленої вище концептуальної моделі метатехнології, у метапроцесі конструювання сервісоорієнтованої технології програмування виділяються шість наступних етапів:

- 1) визначення і фіксація вимог до створюваної СОТП та вибір моделі технологічного процесу розробки ПЗ;
- 2) специфікація вимог до методів і нотацій, як компонент СОТП;
- 3) вибір методів і нотацій, які задовольняють визначеним вимогам;
- 4) специфікація вимог до інструментів – компонентів СОТП;
- 5) пошук і добір потрібних інструментів, які задовольняють визначеним вимогам у веб-середовищі;
- 6) інтеграція компонентів і створення розподіленого середовища розробки ПЗ.

Перший з цих етапів має за кінцеву мету визначення моделі технологічного процесу (ТП), яка буде покладена в основу конструюваної СОТП. Визначення вимог до СОТП проводиться на основі загальних специфікацій ПЗ, яке буде розроблятися за допомогою цієї технології, та специфікацій задач предметної області, яка автоматизується цим ПЗ. Технологічний процес розробки ПЗ визначає дії, їх послідовність, необхідний склад виконавців, засоби і ресурси, і ділиться на сукупність послідовно-паралельних, зв'язаних і супідрядних ланцюжків дій. Кожна з цих дій є ТО, яка переводить розроблюване

ПЗ із деякого вхідного у вихідний (для даної ТО) стан.

Таким чином, технологія програмування задається регламентованою послідовністю ТО, виконуваних на основі того або іншого методу, внаслідок чого стає ясным, не тільки, що має бути зроблене для створення проекту, але як, ким і в якій послідовності. Предметом будь-якої обраної технології програмування слугує відображення взаємозалежних ТО на всіх стадіях життєвого циклу ПС.

Етапи 2, 4 і 3, 5 на перший погляд є тотожними, але все ж таки відрізняються для методів і нотацій з одного боку, і для інструментів – з іншого. Це пов'язано з тим, що інструменти, що включаються до технології як Tool at a Service (TaaS), мають, по-перше, задовольняти специфічним вимогам якості, по-друге, автоматизувати застосування вибраних методів і нотацій. Інакше кажучи, методи і нотації є первинними щодо інструментів і накладають додаткові вимоги до останніх. Саме тому вимоги до методів розробки і нотацій, якими фіксуються отримані артефакти розробки, визначаються на окремому – другому етапі.

На третьому етапі виконується добір методів і нотацій, як компонентів СОТП, виходячи з вимог до них. Також виконується прив'язка вибраних компонентів до етапів і операцій технологічного процесу розробки ПЗ. При цьому враховується сумісність цих компонентів, яка, щоправда, може встановлюватись апріорі (метод використовує певну нотацію), коли на одному етапі застосовується тільки один метод. Якщо ж на різних операціях або послідовностях операції використовуються різні методи, то конструктор СОТП має враховувати їхню сумісність за нотаціями входу та виходу.

Четвертий етап – визначення вимог до інструментів має за мету визначення фактично двох груп вимог. Перша – це вимоги сумісності, які полягають у забезпеченні автоматизованої підтримки інструментом раніше вибраних методів і нотацій. Друга група містить вимоги якості, яким має відповідати інструмент, щоб бути використаним як TaaS у “хмарній”

інфраструктурі, яка є базовою для розподіленого середовища розробки ПЗ.

П'ятий етап – пошук і добір потрібних інструментів у веб-середовищі є найбільш важливим і трудомістким у метапроцесі конструювання СОТП. Основою для нього є дві групи вимог до TaaS, визначені на попередньому етапі й представлені у вигляді онтологічних описів. Пошук інструментів, представлених як веб-сервіси за концепцією TaaS, виконується у веб-середовищі за технологією UDDI (Universal Description, Discovery and Integration). Ця технологія передбачає ведення реєстру веб-сервісів. Підключившись до цього реєстру, конструктор добирає сервіси, які найкращим чином задовольняють визначеним вимогам. Технологія UDDI надає можливість пошуку і публікації потрібного сервісу, при цьому ці операції можуть виконуватися як конструктором, так й іншим веб-сервісом або спеціальною програмою клієнтом.

І на завершальному етапі інтеграції компонентів і створення розподіленого середовища розробки ПЗ відбувається прив'язка компонентів СОТП до етапів і операцій процесу розробки ПЗ. Вона проводиться з урахуванням місця знаходження інструментів на технологічних платформах у розподіленому середовищі розробки. Таким чином при переході від традиційної технології програмування до сервісно-орієнтованої моделі СОТП доповнюється п'ятим компонентом – множиною платформ, на яких розміщена множина інструментів, що включені до СОТП.

Таким чином привнесення в метапроцес конструювання СОТП онтологічного підходу та представлення компонентів створюваної СОТП як веб-сервісів означає, що за допомогою онтологій специфікуються вимоги до створюваної технології та її компонентів. А для представлених у веб-середовищі множин компонентів СОТП створюються їх онтологічні описи. Потім шляхом співставлення виконується пошук і добір потрібних компонентів, з яких формується потрібна технологія розробки ПЗ. Схематично це показано на рис. 2.

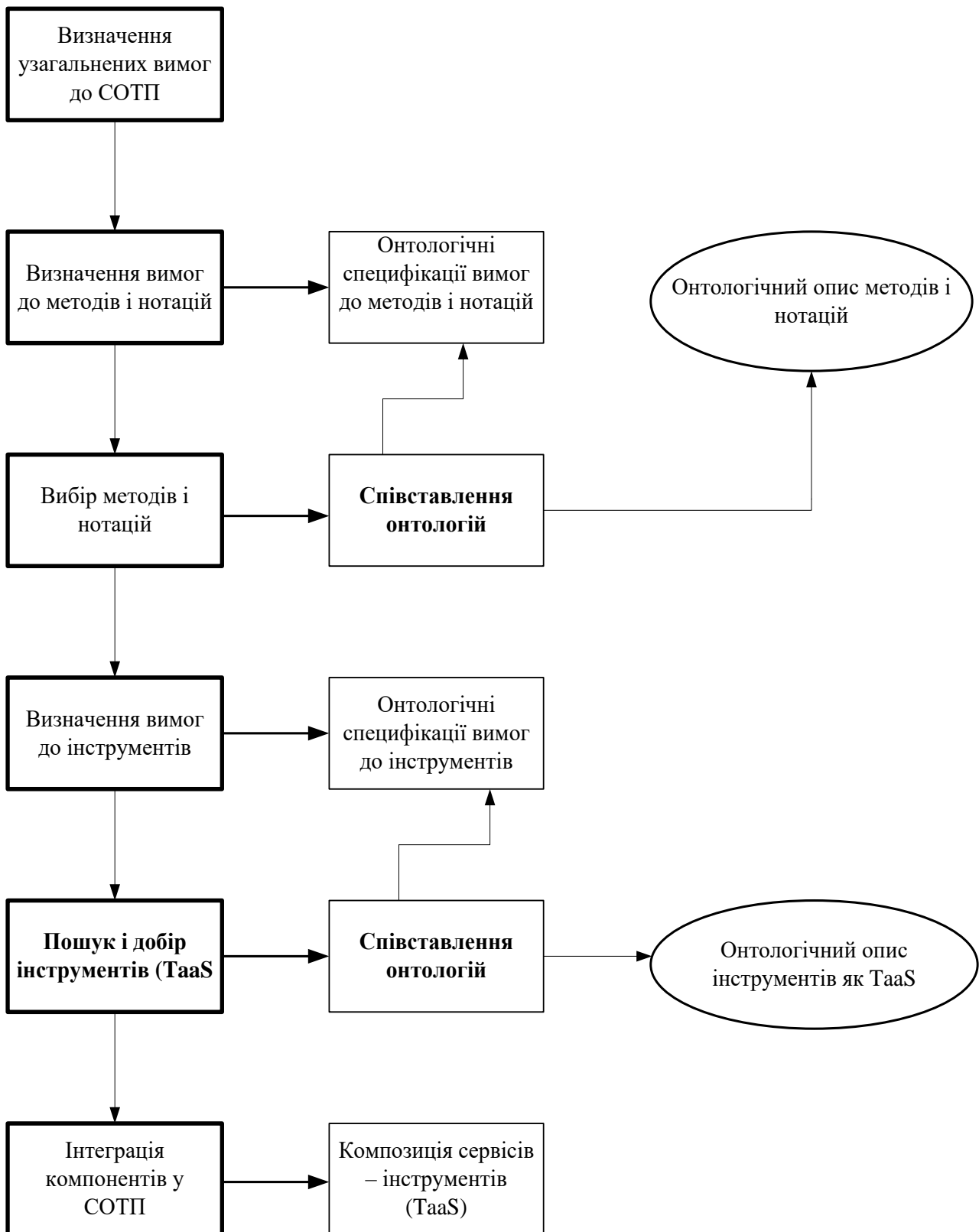


Рис. 2. Модель метатехнологічного процесу конструювання СОПІ



## Архітектура розподіленого середовища розробки програмного забезпечення

Концепція побудови СОТП передбачає її функціонування у розподіленому середовищі розробки ПЗ. Це викликано тим, що самі інструменти розробки, представлені як TaaS, розміщені у веб-середовищі. В основі архітектури розподіленого середовища розробки ПЗ лежить парадигма глобальної розробки програмного забезпечення – Global Software Development (GSD).

Парадигма глобальної розробки ПЗ – Global Software Development (GSD) [4] – полягає у представленні інструментів розробки ПЗ у вигляді веб-сервісів, доступних у веб-середовищі з будь-якої точки планети. При цьому колектив, що займається розробкою проекту може бути територіально розподіленим, тобто його учасники утворюють групи, які можуть географічно перебувати в різних точках і робити свою частину роботи за проектом, співпрацювати з іншими групами через Інтернет-середовище. Кожна група виконує конкретні роботи з створення ПЗ, як правило на певному етапі розробки.

Для виконання цих робіт використовуються відповідні інструменти, і учасникам групи надається доступ до них в Інтернет-середовищі. При цьому інструменти для розробки ПЗ надаються розроблювачам як веб-сервіси по моделі TaaS (Tool as a Service) з використанням «хмарної» технології. Інструменти пов'язані з відповідними групами й надаються тільки членам цієї групи. Інструменти фізично розміщені у веб-середовищі на платформах, які є частиною «хмарної» інфраструктури. Із платформ створюється віртуальне середови-

ще розробки ПЗ, до якого також входять клієнтські термінали розроблювачів і репозиторій артефактів процесу розробки. Інструменти в цьому середовищі взаємодіють як між собою, так і з репозиторієм на підставі прийнятого стандарту. Структура середовища розробки буде менш складною, якщо провайдери, що пропонують інструменти будуть групувати їх на платформах відповідно до функціонального призначення й націленими на підтримку певного етапу процесу розробки. Оскільки колектив розроблювачів також структурується подібним чином, це забезпечить мінімум перехресних посилань типу «розроблювач – платформа». В ідеальному випадку всі інструменти, необхідні певній групі розроблювачів, будуть розміщені на одній платформі.

Застосовуючи даний підхід до СОТП, архітектуру інтегрованого розподіленого середовища розробки ПЗ можна графічно показати на рис. 3.

Структурно воно поділяється на:

- клієнтські термінали розробників ПЗ, які можуть бути географічно розподілені;
- віртуальну технологічну платформу розробки, яка містить інформацію щодо знаходження у мережі включених до СОТП інструментів-сервісів і артефактів проекту розробки ПЗ, а також забезпечує доступ та взаємодію розробників із інструментами та артефактами;
- «хмарний» хостинг інструментів, що входять до СОТП;
- «хмарний» хостинг артефактів які продукуються і використовуються в процесі розробки ПЗ.

*Територіально розподілені групи розробників програмного забезпечення*

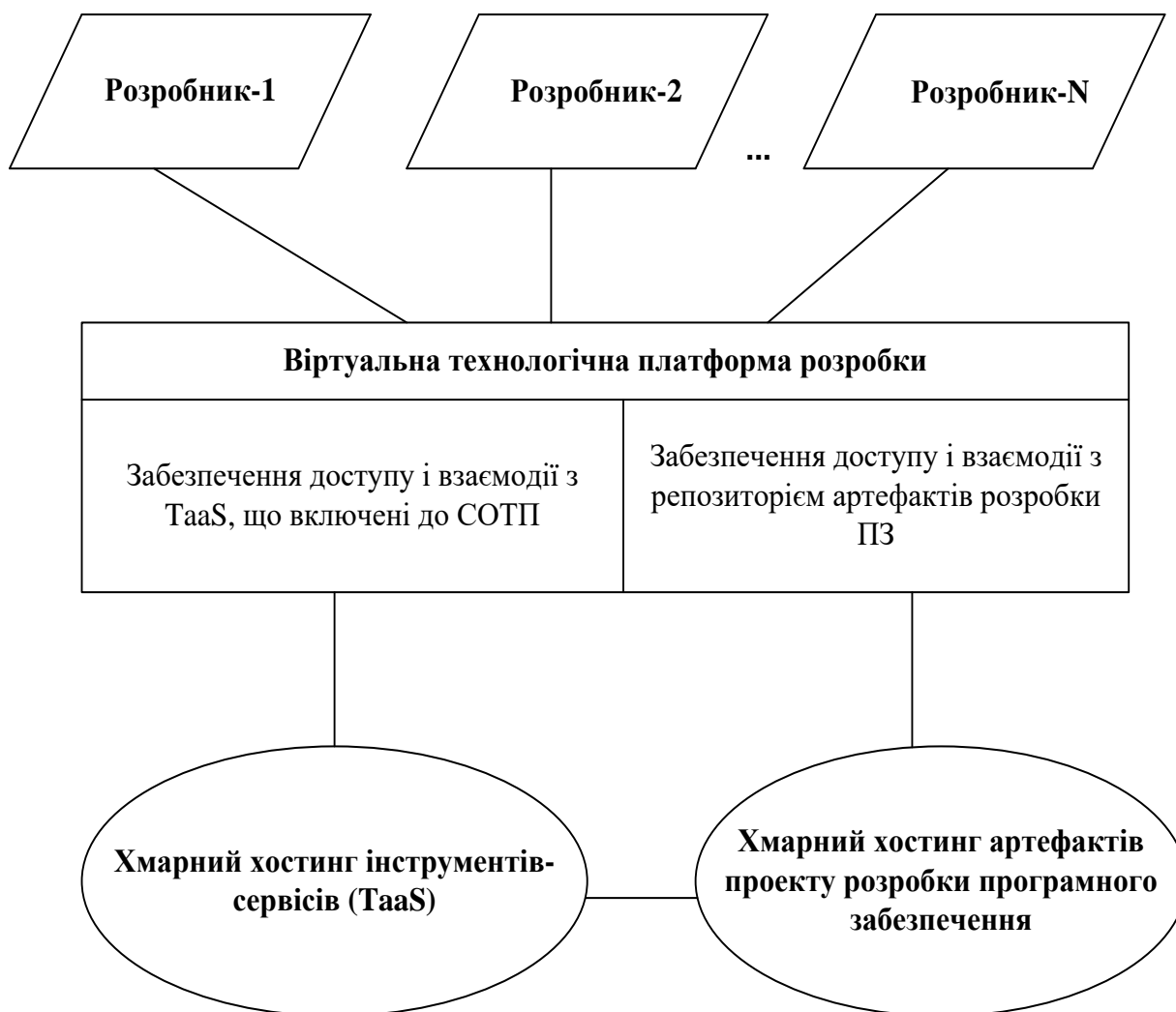


Рис. 3. Архітектура інтегрованого розподіленого середовища розробки ПЗ

**Висновки**

Представлена концепція конструювання СОТП із застосуванням онтології для семантичних описів компонентів технології лежить в основі методології інтелектуалізованого створення СОТП. Сервісно-орієнтований підхід у поєднанні з хмарними технологіями дозволяє використовувати СОТП у розподіленому середовищі розробки ПЗ.

1. *Зінкович В.М., Моренцов Є.І.* Концепції та засоби метатехнології для створення проблемно-орієнтованих технологій програмування. *Проблеми програмування*. 2002. № 3-4. С. 74–82.
2. *McGuinness, D.L. and Harmelen F.V.* OWL Web Ontology Language Overview. 2004. [Електронний ресурс]. Режим доступу: <http://www.w3.org/TR/owl-features/>.
3. Introduction to OWL. 2006. [Електронний ресурс]. – Режим доступу: [http://www.w3schools.com/rdf/rdf\\_owl.asp](http://www.w3schools.com/rdf/rdf_owl.asp).
4. *Hashmi S.I. et al.* Using the Cloud to Facilitate Global Software Development Challenges. *Global Software*

*Engineering Workshop (ICGSEW), Sixth IEEE International Conference. 2011. P. 70–77.*

### **References**

1. Zinkovich V.M., Morentsov Y.I. (2002) Concepts and means of a metatechnology for creation problem-oriented software technologies. Problems in programming. (3-4). P. 74–82. (in Ukrainian).
2. McGuinness, D.L. and F.V. Harmelen. OWL Web Ontology Language Overview. 2004. [Electronic resource]. Mode of access: <http://www.w3.org/TR/owl-features/>.
3. Introduction to OWL. 2006. [Electronic resource]. – Mode of access: [http://www.w3schools.com/rdf/rdf\\_owl.asp](http://www.w3schools.com/rdf/rdf_owl.asp).
4. Hashmi S.I. et al. Using the Cloud to Facilitate Global Software Development Challenges // Global Software Engineering Workshop (ICGSEW), Sixth IEEE International Conference. 2011. P. 70–77.

### **Про автора:**

*Моренцов Євген Іванович,*  
кандидат технічних наук,  
старший науковий співробітник.  
Кількість наукових публікацій в  
українських виданнях – 40.  
<http://orcid.org/0000-0002-0497-11838>.

### **Місце роботи автора:**

Інститут програмних систем  
НАН України,  
03187, м. Київ,  
проспект Академіка Глушкова, 40.  
Тел.: 526 4286.  
E-mail: [yevhen18@diawest.net.ua](mailto:yevhen18@diawest.net.ua)

Одержано 07.02.2017