



Жигаревич О. К.

ДОСЛІДЖЕННЯ МЕТОДІВ ПОБУДОВИ МОДЕЛІ ЕКОСИСТЕМИ НАВЧАЛЬНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У статті запропоновано методи побудови моделей навчального програмного забезпечення. Досліджено сучасне програмне забезпечення, яке характеризується масивністю, складністю. Побудовано метрологічну модель екосистеми навчального програмного забезпечення. Досліджено UML модель екосистеми навчального програмного забезпечення, за допомогою діаграми прецедентів. Розроблено модель екосистеми навчального програмного забезпечення за допомогою мережі Петрі.

Ключові слова: екосистема, навчальне програмне забезпечення, дослідження, SR модель, моделювання, кортеж, мережі Петрі, UML.

1. Вступ

Протягом останніх 10 років загальний стан програмного забезпечення можна охарактеризувати наступними показниками [1]: обсяг ринку — 157 млрд. євро, число підприємств — близько 1,1 мільйонів (США — 39,5 %, Європа — 35,5 %, Азія — 20 %, Африка і Близький Схід — 5 %); обсяг щорічно коду, що розроблюється, — 300 мільйонів рядків; загальний обсяг працюючого коду — близько 1 трильйона рядків. У зв'язку з цим ПЗ набуло важливого потенціалу. Сукупність сучасного ПЗ характеризується глобальним поширенням, колосальними обсягом і складністю, тісною і складною взаємодією систем, широкою інтеграцією в соціальні відносини, появою нових видів взаємодій при розробці, супроводі, обміні, розповсюдженню та використанні. Цей його новий стан мотивує до необхідності використання додаткових понять та концепцій для вивчення та опису. Одним з напрямів такого пошуку є спроби використання концепцій біологічних екосистем.

Дослідження та побудова моделей екосистем навчального програмного забезпечення на даний час є актуальним. Швидкий розвиток інформаційних технологій спонукає до систематизації навчального програмного забезпечення, яке можна регулювати на всіх стадіях життєвого циклу програмного забезпечення. Правильно організована робота розробників, замовників, та користувачів навчального програмного забезпечення — запорука успіху навчання у нашій державі Україні. Екосистема навчального програмного забезпечення поєднує в собі програмне забезпечення, та середовище його використання.

2. Аналіз літературних джерел та постановка проблеми

Поняття та означення «екосистема програмного забезпечення» та супутні терміни активно використовуються виробниками і дослідниками ПЗ. Огляд web-сайтів провідних виробників ПЗ показує, що більшість з них застосовують поняття «екосистема ПЗ» [1, 2], позначаючи ним системи, що включають підприємство

розробника, його ПЗ і партнерів. Виробники застосовують цей термін «як є», без теоретичного обґрунтування або посилання на відповідні праці. Певний виняток становить корпорація Microsoft, яка дає визначення екосистеми ПЗ як сукупності взаємодій і взаємних впливів організацій (державних, навчальних і комерційних) і індивідуумів, що працюють з ПЗ [2]. Microsoft розглядає індустрію ПЗ як цілісну екосистему, в якій повинні існувати ніші для різноманітних видів ПЗ.

Наукові дослідження, що використовують поняття екосистем, на сьогоднішній день представлені кількома працями [2, 3]. У роботі [3] автори описують типові елементи екосистем та їх контекст, а саме:

- ПЗ та його роль в інформаційних технологіях;
- користувачів ПЗ;
- процеси створення ПЗ;
- управління процесами створення та супроводу ПЗ;
- постачання і підтримка ПЗ;
- зв'язок галузі з державними органами;
- економіка ПЗ.

Давши опис перерахованих вище елементів глобальної екосистеми ПЗ, автори намагаються на якісному рівні спрогнозувати характеристики її розвитку з точки зору подальшого підвищення ефективності технологій розробки, появи і розвитку нових галузей застосування. При цьому в роботі не даються ніякі визначення або аналіз терміну «екосистема програмного забезпечення».

У роботі [4] екосистеми розглядаються як рівень абстракції над проектами, який може бути описаний шляхом аналізу нижніх рівнів. Виділяються екосистеми ПЗ як колекції проектів організації, які розробляються і розвиваються в одному середовищі. Автор пропонує метод та засоби для зворотної інженерії екосистем, суть якої полягає в аналізі інформації компонентів проектів для отримання надпроектних високорівневих уявлень, які характеризують організацію, ПЗ та обумовлену соціальну структуру. При цьому, екосистема визначається, як колекція проектів, що представляють суттєві активи організації. У роботі пропонується два погляди на екосистему ПЗ — проектно-орієнтований, що представляє

екосистему з точки зору її проектів і аспектів коду, і орієнтований на розробника, що представляє соціальні аспекти організації. Для створення таких уявлень автор пропонує проводити дослідження проектів організації з точки зору загальних архітектурних аспектів, і безліч розробників організації з точки зору організації типових колективів.

У науковій роботі [4], присвяченій системам ультра-великого масштабу (Ultra-Large-Scale Systems), наголошується, що в галузі існує тенденція до використання концепцій екосистем, для позначення соціально-технічних систем ПЗ. Пояснюють це будовою сучасних систем ПЗ надвеликого масштабу, динамічними взаємозв'язками незалежних і конкуруючих складових у складній системі навколишнього середовища. Під складовими тут розуміють людей, обчислювальні пристрої та організації. З властивостей концепції екосистем, корисних при розгляді великих систем ПЗ, виділяють такі як складність, децентралізоване управління, важко прогнозовані ефекти певних видів, складності моніторингу та оцінювання результатів, змагання в групах, стійкість, адаптивність, стабільність і життєздатність. У звіті відзначається недосконалість аналогії з біологічними екосистемами, однак кажуть про її переваги, оскільки вона найбільш адекватно відображає природу сучасних проблем розробника великих програмних систем, а саме:

- які характеристики можна використовувати для великих програмних систем?
- як проектувати інфраструктуру систем з урахуванням набору сервісів, що забезпечуються учасниками, формальних і соціальних правил регулювання поведінки, інфраструктури ланцюжка поставки?
- як проектувати організаційні процеси, які відповідають неперервному виробництві, та оновлення таких систем?

Таким чином, у розробників і дослідників складається новий погляд на ПЗ як на соціально-технічну систему, що має характеристики, подібні біологічним екосистемам.

Екосистеми характеризуються наступним [5]:

- розмірами (чим вона більше, тим менше залежить від зовнішніх частин);
- інтенсивністю обміну (чим він більший, тим більше приплив і відтік);
- збалансованістю процесів (чим сильніше порушено рівновагу, тим більше повинні бути зміни ззовні для його відновлення);
- стадією розвитку системи.

Для вивчення екосистем біологи використовують два підходи — холистичний, який передбачає вимірювання надходжень та виходу енергії та різних речовин, оцінку сукупних властивостей цілого і, в разі необхідності, вивчення складових частин, і мерологічний, при якому спочатку вивчаються властивості основних частин, а потім ці відомості екстраполюються на систему в цілому.

В Україні відомі роботи науковців [6–9], які протягом останніх 10 років вивчають та досліджують екосистеми програмного забезпечення. Наприклад, у роботі [6] представлено застосування екологічного підходу до дослідження програмного забезпечення. Розглядаються основні положення екології програмного забезпечення, як частини інженерії програмного забезпечення; наводять-

ся три напрямки екологічних досліджень програмного забезпечення («зелене» програмне забезпечення, сталий розвиток, цифрові екосистеми) [6].

Про екосистеми програмного забезпечення відомо [7], де саме досліджується застосування екологічного підходу до програмного забезпечення та формуються основні поняття розділу інженерії програмного забезпечення — екології програмного забезпечення.

Розглянуто використання концепції екосистем виробниками та дослідниками програмного забезпечення, також запропоновано моделі екосистем. Розглянуто придатність та супроводження експлуатації програмного забезпечення авіаційної техніки [8].

3. Об'єкт, мета та задачі дослідження

Об'єктом дослідження є екосистема навчального програмного забезпечення.

Метою роботи є побудова та порівняння моделей екосистеми навчального програмного забезпечення.

Для досягнення поставленої мети були виконані наступні завдання:

- Побудова моделі екосистеми програмного забезпечення.
- Побудова моделей екосистеми навчального програмного забезпечення.
- Створення та дослідження метрологічної моделі екосистеми навчального програмного забезпечення, SR моделі екосистеми навчального програмного забезпечення, Use case діаграми екосистеми навчального програмного забезпечення. Створення та дослідження моделі екосистеми програмного забезпечення за допомогою мережі Петрі.

4. Побудова моделі екосистеми програмного забезпечення

Дослідження біологічних екосистем є організмочентричним, тобто фокусом вивчення є система живих організмів, яка існує та еволюціонує на певному косному біотопі та взаємодіє з живим та косним оточенням на визначених межах екосистеми. При екосистемному підході до дослідження ПЗ центром є програми, які утворюють певну систему та розробляються, супроводжуються та функціонують у певному соціально-технічному середовищі. Соціально-технічне середовище екосистем ПЗ включає апаратне забезпечення, розробників, постачальників, користувачів, регулюючи та стандартизуючи органи та інше.

ПЗ є частиною ІТ і має низку особливих властивостей. Будучи основним компонентом, які реалізують елементи штучного розуму в штучних системах, воно є нематеріальним об'єктом ІТ, і вимагає для свого функціонування і прояву матеріального агента — обчислювальної машини (апаратного забезпечення).

Можна виділити взаємодію ПЗ з іншими елементами екосистеми (рис. 1):

- іншим ПЗ;
- апаратним забезпеченням;
- людиною (користувачем);
- організаціями та соціальними групами;
- штучними об'єктами;
- природними об'єктами та явищами;
- підтримка та оновлення ПЗ.

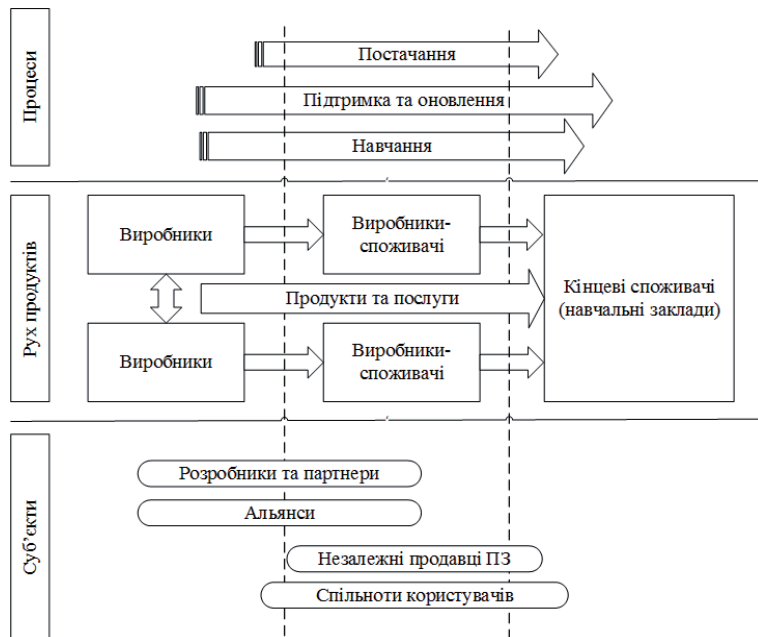


Рис. 1. Модель екосистеми ПЗ

Таким чином, під екосистемою програмного забезпечення будемо розуміти цілісну систему, утворену спільно розроблюваними, супроводжуваними та функціонуючими програмами та взаємодіючу з її соціально-технічним середовищем таким чином, що обмін продуктами та послугами між ними створює чітко визначені структури.

Наведено характеристики, організаційно-технічні системи ПЗ (табл. 1).

До елементів середовищ екосистем ПЗ відносять наступне [9]:

- навчальні заклади та дослідницькі університети, які проводять фундаментальні дослідження та розповсюджують їхні результати;
- компанії, які проводять прикладні дослідження і розробляють комерційні продукти, підвищуючи, таким чином, загальний технологічний рівень;

- держави, що сприяють діяльності інших елементів шляхом захисту прав на інтелектуальну власність;
- органи стандартизації, створюють спільну нормативну базу для взаємодії і розвитку програмних систем.

5. Аналітичний опис моделі екосистеми навчального програмного забезпечення

Спираючись на алгебру систем, можна подати екосистему НПЗ як результат алгебраїчної операції кон'юнкції над автономними складовими системами:

$$ES = \prod_{i=1}^n S_i, n = 5,$$

де ES – екосистема НПЗ; S_i – системи, що складають екосистему НПЗ; \prod – символ операції кон'юнкції над системами.

Оскільки елементи кон'юнкції ПЗ є відкритими системами, будемо подавати їх у вигляді кортежу:

$$S_{ij} = (C, R^C, R^{in}, R^0, B, \Omega, \Theta),$$

де C – непорожня множина компонентів системи, $C = \{c_1, c_2, \dots, c_n\}$; $R^C \subseteq C \times C$ – множина внутрішніх відношень; $R^i \subseteq C_\Theta \times C$ – множина зовнішніх вхідних відношень; $R^0 \subseteq C \times C_\Theta$ – множина внутрішніх вихідних відношень; B – множина поведінки (або функції) $B = \{b_1, b_2, \dots, b_p\}$; Ω – множина структур на компонентах, умови відношень, рамки (область дії) поведінки, $\Omega = \{\omega_1, \omega_2, \dots, \omega_q\}$; Θ – зовнішнє середовище з непорожньою множиною компонентів:

$$C_\Theta, C_\Theta \cap C = \emptyset.$$

Таблиця 1

Загальні характеристики екосистем

Характеристика		
Біологічних систем	Організаційно-технічні системи ПЗ	Інваріантне
Наявність співтовариства організмів	Безліч взаємодіючих елементів з власною поведінкою	Сукупність щодо незалежних продуктів і колективів
Розглядається у встановлених дослідником межах	Наявність встановлених кордонів	Наявність межі програмних систем і організацій
Можливість появи елементів системи із зовні	Спонтанне поява елементів із зовні	Децентралізовані процеси інтеграції та спільного використання різних продуктів
Створення біотичних надорганізменних структур	Спонтанне утворення структур верхнього рівня	Утворення структур над безліччю ПП
Кругообіг речовин і енергії	Обмін елементами	Обмін ПЗ і міграція персоналу
Відкриті системи — отримують і віддають енергію та елементи живлення через середовища на вході і на виході	Імпорт та експорт енергії і речовини	Імпорт і експорт ПП і персоналу
Наявність інформаційної мережі, що управляє екосистемою як єдиним цілим	Децентралізоване управління (саморегуляція)	Тенденція до децентралізації управління (саморегуляції)
Еволюційний розвиток	Еволюція системи	Еволюція
Саморегулюються і адаптуються до змін	Саморегуляція і здатність до адаптації до змінами	Самоорганізація і адаптованість

Абстрактна метрологічна модель екосистеми ПЗ представлена на рис. 2.

Враховуючи введені позначення, можна переписати рівність у вигляді:

$$ES = \prod_{i=1}^n S_i \cong ES \left(U_{i=1}^5 C_i, U_{i=1}^5 R_i^c U_j \Delta R_j^c, U_{j=1}^5 R_j^c, U_{i=1}^5 R_i^0, U_{i=1}^5 B_i U_j \Delta B_j, \Omega_i, U_{i=1}^5 \theta_i \right),$$

де U – символ операції кон'юнкція множини; $\Delta R^c, \Delta B$ – нові внутрішні відношення та емергентна поведінка екосистеми, що виникає при кон'юнкції систем-складових;

$$j = C_n^2 = \frac{n!}{2!(n-2)!},$$

де n – кількість систем, що входить в екосистему НПЗ; 2 – формальна модель екосистеми НПЗ.

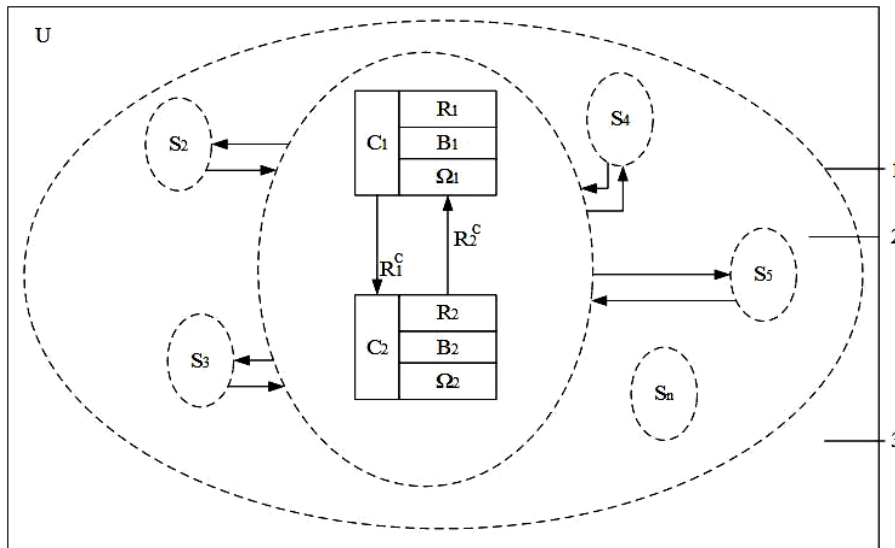


Рис. 2. Метрологічна модель екосистеми навчального програмного забезпечення

Метрологічна модель екосистеми навчального програмного забезпечення містить в собі наступні складові: 1 – межі екосистеми; 2 – соціально-технічне середовище, яке містить автономні елементи системи систем; 3 – зовнішнє середовище, інші екосистеми НПЗ.

Формально описати кожен елемент (множину) кортежа пропонується за допомогою наступних теорій:

- для C, Θ – теорія множин та алгебраїчних операцій над ними;
- для R – теорія відношень. При цьому обов'язково необхідно перевірити виконання властивостей асиметричності та рефлексивності (у випадку невиконання хоча б однієї з властивостей кортеж не можна вважати системою);
- для B, Ω – логіка висловлювань та алгебра процесів реального часу (RTPA). Необхідно враховувати, що поведінка програмної системи поділяється на статичну та динамічну.

Загальні рекомендації з формального опису системи-складових екосистем НПЗ.

Система S_1 «Навчальне програмне забезпечення»:

$$S_1 = \{c_1, R_1^c, R_1^i, R_1^0, B_1, \Omega_1, \Theta_1\}.$$

Відкрита абстрактна система:

$$C_1 = \left\{ c_{ij} \mid c_{ij} \in \bigcup_{j=1}^m P_j, i = \overline{1, n}, j = \overline{1, m}, m, n \in N \right\},$$

де P_j – множина підсистем програмного забезпечення; c_{ij} – компоненти підсистем.

Якщо розглядати ПЗ в термінах діаграми компонентів мови UML, то можна навести наступний приклад.

Приклад опису множини C_1 . Нехай описуване НПЗ – це деяка система, призначена для збору інформації у вигляді відповідей експертів та/або користувачів з подальшою її обробкою. Тоді множина компонентів системи може мати наступний вигляд:

$$C_1 = \left\{ C_{ij} \mid C_{ij} \in \bigcup_{j=1}^3 P_j, i, j \in N \right\},$$

де P_1 – підсистема «Web-модуль»; P_2 – підсистема «База даних»; P_3 – підсистема «Основний додаток».

$c_{i1} = \{\text{файли *.jsp, Web-браузер, TomCat, WebServer}\}$, $i = 1, 7$.
 $c_{i2} = \{\text{таблиці}\}$, $i = 1, 18$.

$c_{i3} = \{\text{бібліотеки, виконуваний компонент}\}$, $i = 1, 16$.

Загальна кількість компонентів системи – 41.

В даному прикладі компонент є фізичною абстракцією, що матеріалізує клас (логічну абстракцію). Можливість більш детального розгляду структури НПЗ з точки зору рівнів інкапсуляції розроблено та представлено результати досліджень [9].

В кожному окремому випадку поняття «компоненту програмного забезпечення» може відрізнитись залежно від потреб дослідників.

$$R_i^c = \{ \rightarrow \},$$

де \rightarrow – відношення базової залежності:

$$(\forall c_{ij} \in P_i, i = \overline{1, n}, j = \overline{1, m}, i, j \in N),$$


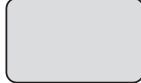
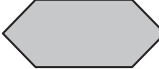






де ψ, ϕ – перетворення «зміна компоненту».

Для того, щоб розглядувана структура (кортеж) справді була системою, необхідно перевірити виконання властивостей рефлексивності та асиметричності для визначеного відношення є очевидною. Властивість асиметричності також виконується, оскільки:

$$(\forall c_{11} \in P_1)(\forall c_{22} \in P_2)(c_{11} \neq c_{22})[c_{11} \rightarrow c_{22} \Rightarrow c_{22} \rightarrow c_{11}].$$

Таблиця 2

Елементи SR моделі

№ п/п	Графічне призначення	Значення
1		Актор
2		Мета
3		Завдання
4		Ресурс
5		Мета (задоволення чуттєвих потреб)
6		Актор з межею
7		Зв'язок залежності
8		Зв'язок декомпозиції
9		Зв'язок значення

$R_1^i = \{ \text{інтеграція (повторно використувані компоненти НПЗ, компоненти ПЗ)} \};$

- розробка та супровід (розробники, компоненти НПЗ);
- функціонування (апаратне забезпечення, компоненти ПЗ);
- створення нормативної бази (органи стандартизації, компоненти ПЗ);
- захист прав на інтелектуальну власність (держава, компоненти ПЗ)}.

$R_1^0 = \{ \text{експорт (компоненти ПЗ, інше ПЗ)} \};$

- функціонування (компоненти ПЗ, апаратне забезпечення);
- надання рішень і послуг (компоненти ПЗ, користувачі);
- постачання (компоненти ПЗ, незалежні продавці ПЗ);
- вплив на навколишнє середовище (компоненти ПЗ, природні явища)}.

$\Theta_1 = \{ \text{апаратне забезпечення, природні об'єкти та явища, соціальна система, економічна система, інші екосистеми ПЗ} \};$

$V_1 = \{ \text{множина поведінок (функцій) системи} \}.$

Як уже зазначалось, поведінка програмної системи поділяються на статичну та динамічну та зручно описується за допомогою алгебри процесів реально-го часу.

6. Програмні засоби створення моделі екосистеми навчального програмного забезпечення

Екосистема навчального програмного забезпечення — це штучний комплекс, який включає програмне забезпечення, середовище його розробки, експлуатації, супроводу та утилізації, які пов'язані між собою обміном програмними продуктами та інтелектом.

Основними елементами екосистеми навчального програмного забезпечення є прикладні продукти та послуги, виробники продуктів та послуг, споживачі, зв'язки. Досліджуються екосистеми програмного забезпечення шляхом створення їх моделей.

Для опису моделей екосистем програмного забезпечення застосовують наступні засоби моделювання і*, UML, мережі Петрі [10].

І* мова моделювання. Використовується для розуміння та моделювання екосистем навчального програмного забезпечення [10]. І* дає можливість розробникам отримати потрібну інформацію на ранньому етапі процесу розробки програмного забезпечення. Засоби І* моделювання мають два основних типи моделей.

Перший тип називається моделлю стратегічної залежності (SD). Вона описує мережу залежностей відношень між учасниками (акторами) системи.

Другий тип, називається обґрунтованою стратегічною моделлю (SR). Вона передбачає представлення того, як кожен актор досягає своєї мети. Модель SR включає в себе SD моделі. Елементами SR моделі є цілі, завдання, ресурси. Отже, елементи SR моделі можна подати у вигляді табл. 2.

Отже, використовуючи нотацію і*, приклад SR моделі екосистеми програмного забезпечення наведено на рис. 3.

Мова UML. Мова UML розвивається з 1994 року і є результатом злиття трьох найбільш відомих об'єктно-орієнтованих підходів: методу Буча, OTM, і OOSE. Вона об'єднує велику кількість різних графічних нотацій з метою впорядкування хаотичного набору графічних засобів, які використовуються під час створення програмного забезпечення. Виступає універсальною мовою моделювання [10]. UML модель екосистеми навчального програмного забезпечення можна представити за допомогою діаграми прецедентів (рис. 4). Діаграма прецедентів (Use Case Diagram) є графічним засобом специфікування вимог, які використовуються для визначення наступного:

- загальних меж та контексту предметної області;
- загальних вимог до функціональної поведінки системи, що проектується;
- взаємодії системи із зовнішнім світом;
- загальні правила поведінки системи у навчальних закладах.

Мережі Петрі. На практиці моделювання систем часто доводиться вирішувати завдання, пов'язані з формалізованим описом і аналізом причинно-наслідкових зв'язків в складних системах, де одночасно паралельно протікає кілька процесів. Мережі Петрі є найпоширенішим в даний час формалізмом, що описує структуру і взаємодію паралельних систем і процесів [10–12].

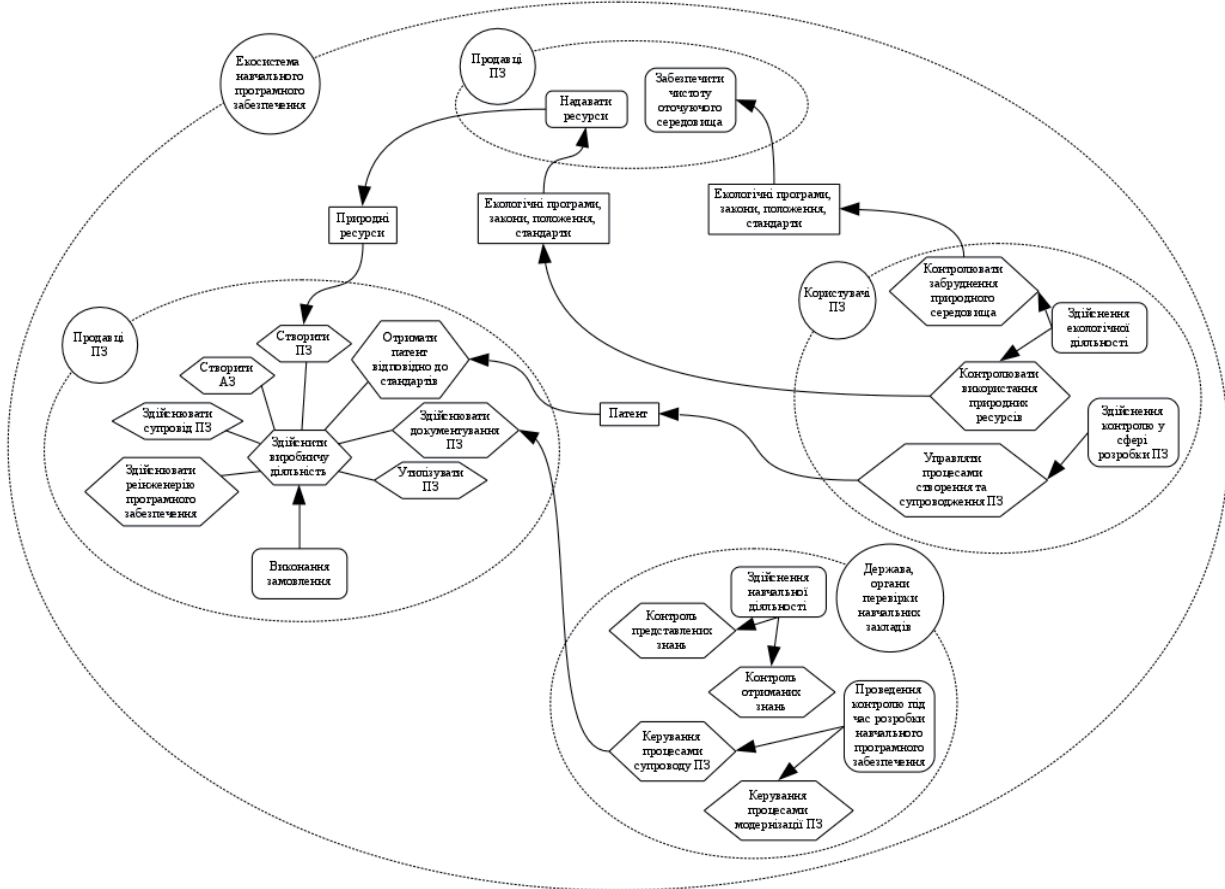


Рис. 3. SR модель екосистеми навчального програмного забезпечення

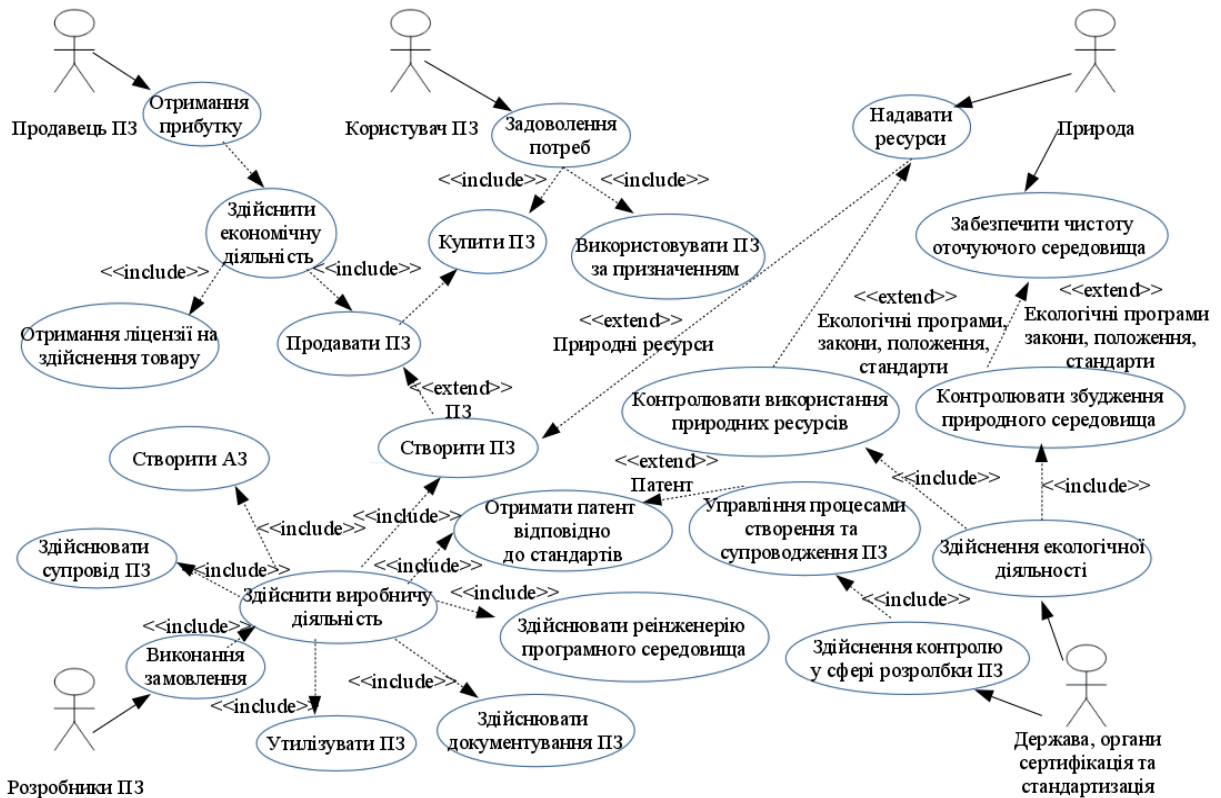


Рис. 4. Use case діаграма екосистеми навчального програмного забезпечення

Формально мережа Петрі задається четвіркою виду:

$$G(V, E, I, O),$$

де V – множина позицій; E – множина переходів; I – вхідні дані функції (пряма функція інцидентності); $I: V \times E \rightarrow \{0, 1\}$; O – вихідні дані функції (обернена функція інцидентності); $O: E \times V \rightarrow \{0, 1\}$.

$$I(e_{\perp j}) = \{v_{\perp i} \in V \mid I(v_{\perp i}, e_{\perp j}) = 1\},$$

$$O(e_{\perp j}) = \{v_{\perp i} \in V \mid O(e_{\perp j}, v_{\perp i}) = 1\},$$

$$I(v_{\perp i}) = \{e_{\perp j} \in E \mid I(e_{\perp j}, v_{\perp i}) = 1\},$$

$$O(v_{\perp i}) = \{e_{\perp j} \in E \mid O(v_{\perp i}, e_{\perp j}) = 1\},$$

$$i = 1, \dots, n, \quad j = 1, \dots, m, \quad n = |V|, m = |E|.$$

Для даної системи її мета та завдання будуть представлятися множиною позицій, а зв'язки між об'єктами будуть множиною переходів.

Графічно модель системи зображується у вигляді дводольних орієнтованих мультіграфів (так як він припускає існування кратних дуг від однієї вершини до іншої), що являє собою сукупність позицій і переходів (рис. 5).

Аналітичне представлення системи за допомогою мережі Петрі матиме наступний вигляд:

$$V = \{n_i, d_j, r_s, p_q, k_t \mid i = 1, 2; j = 1, 5; s = 1, 10; q = 1, 4; t = 1, 3\};$$

$$m = \overline{1, 6};$$

$$I(t_1) = \{n_1\}; \quad O(t_1) = \{r_4\};$$

$$I(t_2) = \{1\}; \quad O(t_2) = \{d_2, d_3\};$$

$$I(t_3) = \{d_2\}; \quad O(t_3) = \{n_2\};$$

$$I(t_4) = \{d_3\}; \quad O(t_4) = \{n_1\};$$

$$I(t_5) = \{d_4\}; \quad O(t_5) = \{d_5\};$$

$$I(t_6) = \{d_5\}; \quad O(t_6) = \{r_3\};$$

$$I(t_7) = \{r_1\}; \quad O(t_7) = \{r_2\};$$

$$I(t_8) = \{r_2\}; \quad O(t_8) = \{r_3, r_4, r_5, r_6, r_7, r_8, r_9, r_{10}\};$$

$$I(t_9) = \{r_3\}; \quad O(t_9) = \{p_3\};$$

$$I(t_{10}) = \{p_1\}; \quad O(t_{10}) = \{p_2\};$$

$$I(t_{11}) = \{p_2\}; \quad O(t_{11}) = \{p_3, p_4\};$$

$$I(t_{12}) = \{p_3\}; \quad O(t_{12}) = \{k_2\};$$

$$I(t_{13}) = \{k_1\}; \quad O(t_{13}) = \{k_2, k_3\}.$$

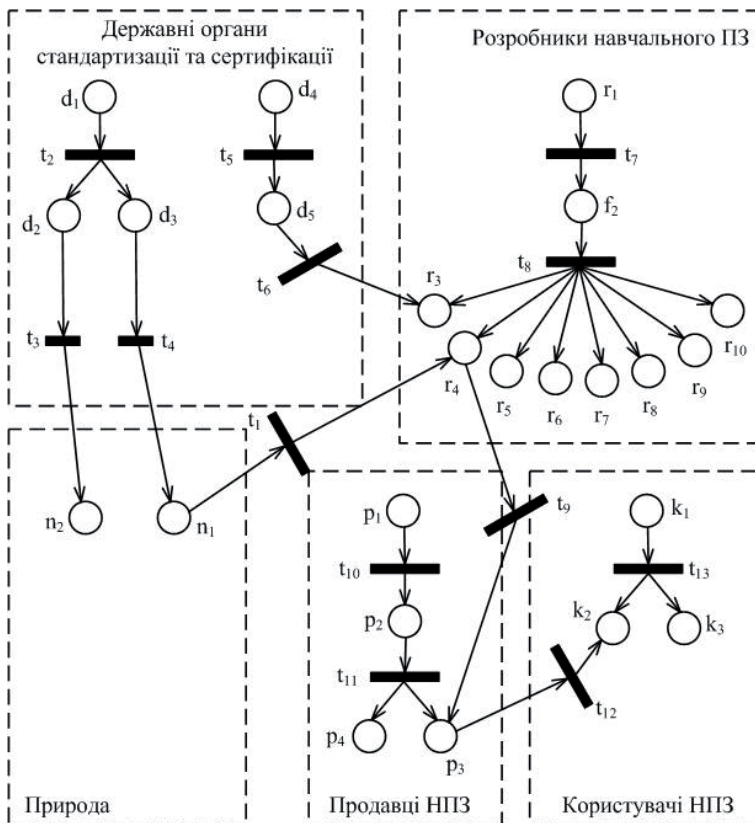


Рис. 5. Модель екосистеми навчального програмного забезпечення за допомогою мережі Петрі

Мережі Петрі застосовуються для опису подій системи довільної тривалості. У цьому випадку модель, побудована з використанням мережі Петрі, відображає тільки порядок настання подій в досліджуваній системі. Для відображення часових параметрів процесу функціонування системи, що моделюється за допомогою мережі Петрі, використовується розширення апарату мереж Петрі: часові мережі, E – мережі, мережі Мерліна тощо [10–13].

Відповідність між розглянутими засобами моделювання представлено у табл. 3.

В роботах [12–14] екосистема програмного забезпечення розглядається з точки зору концепції системи систем (System of systems). Система систем – це будь-яка система, що складається із систем, які самі по собі є автономними. Під системою розуміється будь-яка взаємодіюча або взаємозалежна група сутностей, яка формує цілеспрямовану єдність. Під автономністю розуміється здатність кожної системи здійснювати самостійні дії або приймати рішення. Оперативна, управлінська та еволюційна незалежність і несподівана поведінка впливають з автономності компонентів системи систем. Усе вищезгадане дає можливість розглядати екосистему навчального програмного забезпечення як систему систем, автономними компонентами якої будемо вважати:

- програмне забезпечення (технічна абстрактна система);
- апаратне забезпечення (технічна матеріальна система);
- природні об'єкти та явища (природна система);
- держава, органи стандартизації, виробники та продавці, користувачі програмного забезпечення (економічна система);
- розробники та менеджери, вищі учбові заклади (соціальна система).

програмного забезпечення, як одну із галузей науки, яку потрібно, ще довго досліджувати.

Література

- Jansen, S. Understanding Software Ecosystems: A Strategic Modeling Approach [Text] / S. Jansen, J. Bosch // Proceedings of the Workshop on Software Ecosystems. — 2011. — P. 65–76.
- Duinkerken, W. Transaction Cost Economics in Software Ecosystems [Text] / W. Duinkerken // Some empirical evidence. — 20 April 2009. — P. 22–37.
- Dyba, T. Empirical studies of agile software development: A systematic review [Text] / T. Dyba, T. Dingsoyr // Information and Software Technology. — 2008. — Vol. 50, № 9–10. — P. 833–859. doi:10.1016/j.infsof.2008.01.006
- Messersmith, D. Ultra-Large-Scale Systems [Text] / D. Messersmith, C. Szyperski // Understanding an Indispensable Technology and Industry. — London: MIT press, 2003. — 233 p.
- Сидоров, М. О. Программное обеспечение — экологический подход к исследованиям [Текст] / М. О. Сидоров // Инженерия программного обеспечения. — 2010. — № 1. — С. 5–13.
- Сидоров, Н. А. Экология программного обеспечения [Текст] / Н. А. Сидоров // Инженерия программного обеспечения. — 2010. — № 1. — С. 53–61.
- Хоменко, В. А. Экосистемы программного обеспечения [Текст] / В. А. Хоменко // Вісник Національного технічного університету «ХПІ». — 2011. — № 23. — С. 114–118.
- Луцький, М. Підтримка придатності та супроводження експлуатації програмного забезпечення авіаційної техніки [Текст] / М. Луцький, М. Сидоров, Ю. Рябокіль // Проблеми програмування. — 2010. — № 23. — С. 229–239.
- Sidorov, N. A. Software Ecosystems Modeling [Text] / N. A. Sidorov, O. O. Grinenko // Engineering Software. — 2013. — № 2(14). — P. 38–48.
- Томашевський, В. М. Моделювання систем [Текст]: підручник / В. М. Томашевський. — Київ, 2005. — 349 с.
- Boucharas, V. Formalizing software ecosystem modeling [Text] / V. Boucharas, S. Jansen, S. Brinkkemper // Proceedings of the 1st international workshop on Open component ecosystems. — 2004. — P. 41–50. doi:10.1145/1595800.1595807
- Jansen, S. A sense of community: A research agenda for software ecosystems [Text] / S. Jansen, A. Finkelstein, S. Brinkkemper // 31th International Conference on Software Engineering, New and Emerging Research Track. — 2009. — P. 187–190. doi:10.1109/icse-companion.2009.5070978
- Lungu, M. F. Reverse Engineering Software Ecosystems [Text]: Doct. Diss. / M. F. Lungu. — Lugano, 2009. — 208 p.
- Webber, L. Green Tech: How to Plan and Implement Sustainable IT Solutions [Text] / L. Webber, M. Wallace. — AMACOM, 2009. — 304 p.

Таблиця 3

Відповідність між засобами моделювання екосистем

№ п/п	Елементи екосистеми	i*	UML	Мережа Петрі
1	актор			—
2	мета			
3	завдання			
4	ресурс		—	—
5	актор з межею		—	—
6	зв'язок залежності		—	
7	зв'язок декомпозиції		<code><<extend>></code> ----->	
8	зв'язок значення		<code><<include>></code> ----->	

7. Висновки

В результаті проведених досліджень:

1. У даній роботі зроблений аналітичний опис моделі екосистеми навчального програмного забезпечення, за використанням алгебри систем. Екосистему навчального програмного забезпечення представлено, як результат алгебраїчної операції кон'юнкції над автономними складовими системами. Побудована метрологічна модель екосистеми навчального програмного забезпечення.

2. Досліджено програмні засоби створення моделі екосистеми навчального програмного забезпечення. Моделі екосистем навчального програмного забезпечення побудовані за допомогою мереж Петрі, Use case діаграми екосистеми навчального програмного забезпечення та SR моделі екосистеми навчального програмного забезпечення.

3. Екосистема навчального програмного забезпечення представляє собою взаємозв'язок розробників програмного забезпечення, користувачів програмного забезпечення, та навчальну установу ВУЗ. Потрібно розуміти, що ключову роль у цій взаємодії відіграє людина, бо вона розробляє, користується, та навчається. Можна говорити про екосистему навчального

ИССЛЕДОВАНИЕ МЕТОДОВ ПОСТРОЕНИЯ МОДЕЛИ ЭКОСИСТЕМЫ УЧЕБНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В статье предложены методы построения моделей учебного программного обеспечения. Исследовано современное программное обеспечение, которое характеризуется массивностью, сложностью. Построена метрологическая модель экосистемы учебного программного обеспечения. Исследована UML модель экосистемы учебного программного обеспечения, с помощью диаграммы прецедентов. Разработана модель экосистемы учебного программного обеспечения с помощью сети Петри.

Ключевые слова: экосистема, учебное программное обеспечение, исследования, SR модель, моделирование, кортеж, сети Петри, UML.

Жигаревич Оксана Костянтинівна, асистент, кафедра комп'ютерної інженерії, Луцький національний технічний університет, Україна, e-mail: oz_lutsk@mail.ru.

Жигаревич Оксана Константиновна, асистент, кафедра комп'ютерної інженерії, Луцький національний технічний університет, Україна.

Zhyharevych Oksana, Lutsk National Technical University, Ukraine, e-mail: oz_lutsk@mail.ru