УДК 004. 413 (045)

О. О. Гріненко, *старший викладач*
Інститут комп'ютерних інформаційних технологій
Національний авіаційний університет
просп. Комарова, 1, корпус 6, м. Київ, Україна
gsa_ck@ukr.net

МОНІТОРИНГ ЕКОСИСТЕМ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У статті розглядається застосування екологічного підходу до досліджень програмного забезпечення та формуються основні поняття розділу інженерії програмного забезпечення – екології програмного забезпечення. Екологія програмного забезпечення – це термін, яким позначається галузь дослідження властивостей, поведінки і законів систем програмного забезпечення та їх впливу на середовище існування і діяльності людини. Ключову роль у дослідженні екології програмного забезпечення відіграють екосистеми програмного забезпечення (ЕПЗ). Під екосистемою програмного забезпечення будемо розуміти комплекс, який включає в себе програмне забезпечення, середовище його розробки, експлуатації, супроводу та утилізації, які пов'язані між собою обміном програмними продуктами та інформацією. Важливим у дослідженні ЕПЗ є їх моніторинг.

Ключові слова: *програмне забезпечення, екосистема програмного забезпечення, моніторинг екосистем програмного забезпечення, характеристики моніторингу екосистем програмного забезпечення.*

Постановка проблеми. Програмне забезпечення стало глобальним явищем, що включає безліч тісно взаємодіючих соціально-технічних систем. Сукупність сучасного програмного забезпечення 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, розповсюдженні та використанні. Таке його становище викликає потребу у використанні додаткових понять і концепцій для вивчення і опису програмного забезпечення. Одним із проявів такої потреби є сучасні спроби використання концепції екосистем програмного забезпечення. Розвиток програмного забезпечення є поступовим ітераційним процесом, кожний цикл якого забезпечує нові можливості. Однак, як і еволюція біологічних видів, еволюція програмного забезпечення не може розглядатися без урахування розвитку його екосистеми, оскільки вимагає відповідного розвитку інших елементів екосистеми – кваліфікації та досвіду розробників і партнерів, навичок і сподівань користувачів. Тому, екосистеми програмного забезпечення можуть стати додатковим інструментом дослідження та оцінювання еволюції програмного забезпечення.

Аналіз останніх досліджень. Наукові дослідження, що використовують поняття екосистем програмного забезпечення, представлені кількома працями. В роботі [12] під екосистемою програмного забезпечення розуміється множина акторів, які функціонують як єдине ціле і взаємодіють з ринком ПЗ та послуг, а також відношення між ними. Ці відношення часто базуються на єдиній технологічній платформі або ринку і співпрацюють на основі обміну інформацією, ресурсами або артефактами. З точки зору ЕПЗ, угоди між постачальниками ПЗ зосереджуються навколо програмних компонентів і послуг. В літературі і серед вчених, здається, не існує єдиного означення поняття «компоненти ПЗ». В роботі [10] автори розглядають компонент ПЗ як пакет функцій ПЗ, які доступні через єдиний інтерфейс, що використовується або може бути використаний як елемент в інших пакетах ПЗ, але функціональне ядро розвивається окремо від цих пакетів. Послуги ПЗ – це компоненти ПЗ, які доступні через зв'язки за межами природного середовища користувачів.

В дослідженні [9] розкривається поняття ЕПЗ через поняття «людської екосистеми». Людська екосистема складається з акторів, зв'язків між ними, їх діяльності і операцій, що стосуються фізичних або нефізичних факторів. Необхідно також розуміти, що людські

екосистеми – це економічні та соціальні системи. Економічні екосистеми характеризуються акторами, які можуть бути підприємствами, постачальниками або клієнтами; факторами, які є продуктами та послугами; а також операціями, що включають в себе фінансові операції, обмін інформацією, знаннями, тощо. Соціальні екосистеми складаються з користувачів, їх соціальних зв'язків та обміну інформацією. ЕПЗ складається з множини програмних рішень, що дозволяють підтримувати і автоматизувати діяльність і операції між акторами та організаціями, що надають ці рішення. ЕПЗ, в першу чергу, є економічними екосистемами, а, отже, товари та послуги є програмними рішеннями та послугами, що надають, забезпечують підтримку та автоматизацію діяльності та операцій.

Корпорація «Майкрософт» дає визначення екосистеми програмного забезпечення як сукупності взаємодій і взаємних впливів організацій (державних, навчальних і комерційних) та індивідуумів, що працюють з програмним забезпеченням [7].

У роботі [13] екосистеми розглядаються як рівень абстракції над програмними продуктами і проектами, який може бути описаний шляхом аналізу нижніх рівнів. Автор пропонує метод і засоби для зворотної інженерії екосистем, суть якої полягає в аналізі інформації компонентів проектів для отримання надпроектних високорівневих уявлень, що характеризують організацію, компоненти програмного забезпечення та обумовлену соціальну структуру.

У роботі [17] «Екосистеми програмного забезпечення» 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 підтримку ПЗ; зв'язок галузі з державними органами; економіку ПЗ.

У роботі [16], присвяченій системам ультравеликого масштабу (Ultra-Large-Scale Systems), підкреслюється, що в галузі існує тенденція до використання концепцій екосистем для позначення соціально-технічних систем ПЗ. Автори пояснюють це подобою сучасних систем ПЗ ультравеликого масштабу динамічним співтовариствам незалежних і конкуруючих організмів у складному змінному середовищі. Під організмами тут розуміються люди, обчислювальні пристрої та організації.

Отже, під екосистемою програмного забезпечення будемо розуміти штучний комплекс, який включає в себе програмне забезпечення, середовище його розробки, експлуатації, супроводу та утилізації, пов'язаних між собою обміном програмними продуктами та інтелектом. Важливим у дослідженні ЕПЗ є їх моніторинг.

Мета. Для вирішення завдання моніторингу екосистем проекту програмного продукту необхідно проаналізувати склад елементів проекту, побудувати модель екосистеми [16], що описує ці елементи, визначити способи побудови моделі екосистеми конкретного програмного продукту на основі різних елементів проекту, визначити підхід до аналізу побудованої моделі для оцінювання якісних характеристик проекту.

Виклад основних результатів. Основними складовими ЕПЗ є саме програмне забезпечення, а також середовище його розробки, основною характеристикою якого є компетенція розробників.

Виходячи з цього, з метою досягнення мети дослідження пропонується методика моніторингу ЕПЗ, що складається з двох етапів:

1. визначення характеристик ЕПЗ, що розробляється;
2. визначення компетенцій персоналу ЕПЗ.

Щоб визначити характеристики ЕПЗ, необхідно розв'язати такі завдання:

1) проаналізувати зв'язок між характеристиками ПЗ і метричними показниками;
2) формалізувати задачу визначення характеристик ПЗ;
3) розробити метод визначення характеристик.

Задачу визначення якісних характеристик можна сформулювати як задачу класифікації досліджуваного програмного продукту до одного з визначених для даної характеристики класів. Оскільки характеристики виражаються лінгвістично, для формалізації задачі пропонується застосувати апарат нечіткої логіки. У цьому випадку можна зіставити кожну характеристику з лінгвістичною змінною, термами якої є можливі класи. Вхідними даними для задачі визначення якісних характеристик екосистем програмного забезпечення є модель проекту програмного забезпечення, що описана в попередньому розділі. Таким чином, задача визначення однієї якісної характеристики програмного забезпечення зводиться до задачі нечіткої класифікації системи, представленої у вигляді семантичної мережі [4]

$$Y = \{y_{CPLX}, y_{VLTL}, y_{VNTN}, y_{WCK}, y_{RCCL}\} = \\ = \begin{cases} F_{CPLX}(MODEL), F_{VLTL}(MODEL), \\ F_{MNTN}(MODEL), F_{WCK}(MODEL), \\ F_{RCCL}(MODEL) \end{cases} \quad (1)$$

де Y – вектор якісних характеристик;

Fx (MODEL) – функція для обчислення характеристики *x*. Однак аналіз системи, представленої семантичним графом, дуже складний, тому пропонується спростити модель, замінивши семантичну мережу на її метричні показники:

$$X = \{M_1(MODEL), \dots, M_N(MODEL)\}$$
$$Y = \{y_{CPLX}, y_{VLTL}, y_{MNTN}, y_{WCK}, y_{RCCL},\} = \\ = \{F_{CPLX}(X), F_{VLTL}(X), F_{MNTN}(X), F_{WCK}(X) = \\ , F_{RCCL}(X)\}, \quad (2)$$

де *X* – вектор метричних показників;

Mi(MODEL) – алгоритм обчислення метрики *i* на основі проектної моделі;

Fxy (X) – функція для обчислення характеристики *y* на основі вектора метричних показників.

Таким чином, задача визначення характеристик екосистеми поділяється на дві підзадачі: обчислення метричних показників семантичного графа моделі системи і нечітку класифікацію системи за обраними характеристиками на основі обчислених метричних показників.

Метод визначення характеристик ЕПЗ, що розробляється, базується на таких основних положеннях:

1. Як вхідна інформація про аналізовану систему використовуються значення метричних показників проектної моделі.

2. Для вхідних змінних (значень метричних показників) визначено діапазон допустимих значень.

3. Вхідні змінні представлені чіткими чисельними значеннями.

4. Вихідні змінні виражені лінгвістично.

5. Задача знаходження значень вихідних змінних за вхідними даними вирішується із застосуванням нечіткої нейронної продукційної мережі.

6. Для нечіткої продукційної мережі експертами проводяться операції формування та адаптації.

Оцінювати компетенції персоналу пропонується на основі оцінювання змін у проекті, які робить кожний учасник команди розробки. Виходячи з твердження, що набір компетенцій персоналу, яким слід приділяти увагу в процесі розробки, повинен точно корелюватися з характеристиками програмного продукту, цей характер змін у проекті є найбільш об'єктивною основою для оцінювання персоналу. У цій роботі розглядаються тільки компетенції, що впливають на якість проектної моделі.

Нехай A_i – елементи проекту з номером версії *i*, B_m – вектор компетенцій учасника команди розробки *m*.

Задачу визначення компетенцій можна представити з використанням такого математичного запису:

$$B_m = F(\{I_m(A_i, A_{i-1})\}), \quad (3)$$

де B_m – вектор компетенції розробника,

F – функція, визначає компетенції розробника на основі векторів впливу вироблених ним змін.

$I_m(A_i, A_{i-1})$ – вектор факторів впливу на проект *i* зміни виробленого розробником *m*.

Функція $F(\{I\})$ може бути обчислена на основі вектора усереднених значень факторів впливу:

$$\bar{I} = \frac{\sum_{i=1}^{N} I_i}{N}. \quad (4)$$

Висновки. Основними складовими ЕПЗ є саме програмне забезпечення, а також середовище його розробки, основною характеристикою якого є компетенція розробників. Для того щоб визначити характеристики екосистеми, необхідно перейти до задачі нечіткої класифікації проектної моделі програмного продукту. Для того щоб знизити складність завдання класифікації і зробити метод визначення характеристик програмного забезпечення екосистем, що розробляється, більш гнучким, пропонується вирішувати завдання визначення характеристик, використовуючи обчислені за моделлю проекту метричні показники. Для вирішення завдання нечіткої класифікації в умовах відсутності великої кількості експериментальних даних застосовується математичний апарат нечітких нейронних мереж.

Список літератури

1. Любимский Э. З. На пути к построению общества программ / Э. З. Любимский // Программирование. – 2009. – № 1. – С. 4–10.

2. Мазур И. И. Курс инженерной экологии / И. И. Мазур, О. И. Молданов. – М. : Высшая школа, 1999. – 447 с.
3. Кондратюк Є. М. Словник-довідник з екології / Є. М. Кондратюк, Г. І. Хархота. – К. : Урожай, 1987. – 160 с.
4. Кривий С. Л. Курс дискретної математики. / С. Л. Кривий. – К. : НАУ-друк, 2010. – 368 с.
5. Луцький М. Г. Методологія інформатизації науково-практичного супроводження експлуатації авіаційної техніки / М. Г. Луцький // Інженерія програмного забезпечення. – Київ, 2011. – № 1. – С. 25–35.
6. Сидоров Н. А. Экология программного обеспечения / Н. А. Сидоров // Інженерія програмного забезпечення. – Київ, 2010. – № 2. – С. 1–9.
7. Советов Б. Я. Моделирование систем / Б. Я. Советов, С. А. Яковлев. – М. : Высшая школа, 2001. – 344 с.
8. Хоменко В. А. Экосистемы программного обеспечения / В. А. Хоменко // Вісник НТУ. – Київ, 2011. – № 23. – С. 114–118.
9. Чернышов Л. Н. Экология программирования / Л. Н. Чернышов. – М., 2008.
10. Bosch, J. (2009) From software product lines to software ecosystems. *13th International software product line conference*. – San Francisco, August 24 – 28, pp. 1–10.
11. Duinkerken, W. (2009) Transaction cost economics in software ecosystems: some empirical evidence. April 20, pp. 22–37.
12. Florina, C. (2000) The digital ecosystem – creating digital devices. *Conf. Proc*., Oct. 16.
13. Jansen, S., Finkelstein, A. and Brinkkemper. (2009) A sense of community: A research agenda for software ecosystems. *31th International Conference on Software Engineering, New and Emerging Research Track*.
14. Lungu, M. F. (2009) Reverse engineering software ecosystems. Doct. Diss. USI, 208 p.
15. Messershmitt, D. G. and Szyperski, C. (2003) Software ecosystems: understanding an indispensable technology and industry. London: MIT press, 233 p.
16. Sidorov M. O. Software ecosystems modeling / M. O. Sidorov, O. O. Grinenko // Інженерія програмного забезпечення. – Київ, 2013. – № 2 (14). – С. 38–48.
17. Schulz, G. (2009) The green and virtual data center. Taylors Francis G, 218 p.
18. Velye, Y., Velye, A. and Elsenpeter, R. (2008) Green IT: Rednee your information systems environmental input while adding to the bottom line.
19. Webber, L. and Wallance M. (2009) Green tech: thow two and to plan implement sustainable IT solutions. FBACOM, 29 p.

References

1. Lyubimskii, E. Z. (2009) On the way to building of program society. *Programmirovaniye*, (1), pp. 4-10 [in Russian].
2. Mazur, I. I. and Moldanov, O. I. (1999) The cource of engineering ecology. Moscow: Vysshaya shkola, 447 p. [in Russian].
3. Kondratyuk, E. M. and Kharhota, G. I. (1987) The dictionary-directory on ecology. Kyiv: Urozhay, 160 p. [in Ukrainian].
4. Kryvyi, S. L. (2010) The course of descrete mathematics. Kyiv: NAU-druk, 368 p. [in Ukrainian].
5. Lutskii, M. G. (2011) The methodology for informatization of scientific and practical maintenance of aeronautical engineering operation. *Inzheneriya programnoho zabezpechennya*. Kyiv, (1), pp. 25–35 [in Ukrainian].
6. Sydorov, N. A. (2010) Software ecology. *Inzheneriya programnoho zabezpechennya*. Kyiv, (2), pp. 1–9 [in Russian].
7. Sovyetov, B. Ja. and Yakovlev, S. A. (2001) Systems modeling. Moskow: Vysshaya shkola, 344 p. [in Russian].
8. Khomenko, V. A. (2011) Software ecosystems. *Visnyk NTU*. Kyiv, (23), pp. 114–118 [in Russian].
9. Chernyshov, L. N. (2008) Programming ecology. Moscow [in Russian].
10. Bosch, J. (2009) From software product lines to software ecosystems. *13th International software product line conference*. – San Francisco, August 24 – 28, pp. 1–10.
11. Duinkerken,, W. (2009) Transaction cost economics in software ecosystems: some empirical evidence. April 20, pp. 22–37.
12. Florina, C. (2000) The digital ecosystem – creating digital devices. *Conf. Proc*., Oct. 16.
13. Jansen, S., Finkelstein, A. and Brinkkemper. (2009) A sense of community: A research agenda for software ecosystems. *31th International Conference on Software Engineering, New and Emerging Research Track*.
14. Lungu, M. F. (2009) Reverse engineering software ecosystems. Doct. Diss. USI, 208 p.
15. Messershmitt, D. G. and Szyperski, C. (2003) Software ecosystems: understanding an

indispensable technology and industry. London: MIT press, 233 p.
16. Sidorov M. O., Grinenko O. O. (2013) Software ecosystems modeling. *Inzheneriya programnoho zabezpechennya*. Kyiv, (2), (14), pp. 38–48.
17. Schulz, G. (2009) The green and virtual data center. Taylors Francis G, 218 p.
18. Velye, Y., Velye, A. and Elsenpeter, R. (2008) Green IT: Rednee your information systems environmental input while adding to the bottom line.
19. Webber, L. and Wallance M. (2009) Green tech: thow two and to plan implement sustainable IT solutions. FBACOM, 29 p

Стаття надійшла до редакції 17.04.2014.

O. O. Grinenko, *senior lecturer*
Institute of Computer Information Technologies National Aviation University
Komarova Ave., 1, building 6, Kyiv, Ukraine
gsa_ck@ukr.net

MONITORING OF SOFTWARE ECOSYSTEMS

In the article the use of ecological approach to the research of software is considered and the basic concepts of software engineering part (software ecology)are formed. Software ecology is a notion that includes the study of properties, behavior and laws of software systems and their impact on human environment. Software ecosystems (SECOs) play a key role in the study of software ecology. Software ecosystem is a complex that includes software and environment of its development, operation, maintenance and disposal which are interconnected by software exchange and information. Monitoring of software ecosystems plays an important role in their research.

Keywords: *software, software ecosystem, monitoring of software ecosystems, metrics of monitoring of software ecosystems.*