УДК 004.413 (045)

ЕКОСИСТЕМА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЯК СИСТЕМА СИСТЕМ

О. О. Гріненко

Національний авіаційний університет

Elena.Grinenko@livenau.net

Розглянуто проблему моделювання екосистем програмного забезпечення, їх формального опису, а також представлення екосистеми програмного забезпечення як системи систем.

Ключові слова: екосистема програмного забезпечення; моделювання; система систем; алгебра процесів у реальному часі.

A problem of software ecosystem modeling and their formal description and presentation of software ecosystem as a system of systems are considered.

Keywords: software ecosystem; modeling; system of systems; real-time process algebra.

Вступ

Дослідження екосистем складається з їх аналізу та побудови представлень. Складність екосистеми програмного забезпечення і різноманітність цілей зацікавлених осіб зумовлює наявність багатьох поглядів, з яких можна подивитися на екосистему. З властивостей концепцій екосистем, корисних при розгляді великих систем програмного забезпечення, виділяють такі, як складність, децентралізоване управління, важко прогнозовані ефекти, складність моніторингу та оцінювання, змагання в нішах, стійкість, адаптивність, стабільність і життєздатність. Ці особливості, а також склад екосистеми програмного забезпечення дають змогу розглянути її з погляду системи систем (System of a System).

Аналіз досліджень

Визначення і формальний опис екосистем програмного забезпечення важливі сьогодні. Дослідження екосистем програмного забезпечення наведено в працях [1], [3], [4], [5], [6], [10], [11], [12].

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

Основними елементами екосистеми програмного забезпечення є продукти та послуги, виробники продуктів та послуг, споживачі, зв'язки. Важливим у дослідженні екосистем програмного забезпечення є створення їх моделей, а також засобів для моделювання екосистем програмного забезпечення та дослідження основних задач, пов'язаних з моделюванням екосистем програмного забезпечення.

Постановка завдання

Система систем — це будь-яка система, що складається із автономних систем. Під системою маємо на увазі будь-яку взаємодіючу або взаємозалежну групу сутностей, які формують цілеспрямовану єдність. Під автономністю розуміється здатність здійснювати самостійні дії або прийняття рішень. Оперативна, управлінська та еволюційна незалежність і емергентна поведінка випливають з автономності компонентів системи систем [6]. Усе вищезгадане дає можливість розглядати екосистему програмного забезпечення як систему систем, автономними компонентами якої будемо вважати:

1. Програмне забезпечення (технічна абстрактна система S_1).

2. Апаратне забезпечення (технічна матеріальна система S_2).

3. Природні об'єкти та явища (природна система S_3).

4. Держава, органи стандартизації, виробники та продавці, користувачі ПЗ (економічна система S_4).

5. Розробники та менеджери, соціальні спільноти (соціальна система S_5).

Отже, під екосистемою програмного забезпечення будемо розуміти систему систем, утворену взаємодією автономних відкритих систем технічного, природного, соціального та економічного походження з метою забезпечення визначеної продуктивності. Залежно від конкретного аспекту, який цікавить дослідника, автономні компоненти можуть бути доповнені або уточнені.

Виклад основного матеріалу

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

$$ES = \coprod_{i=1}^{n} S_i,\ n = 5, \qquad (1)$$

де ES — екосистема ПЗ; S_i — системи, що складають екосистему ПЗ; $\coprod$ — символ операції кон'юнкції над системами.

Оскільки елементи екосистеми ПЗ є відкритими системами, будемо подавати їх у вигляді кортежу:

$$S = (C, R^C, R^i, R^O, B, \Omega, \Theta),$$

де C — непорожня множина компонентів системи, $C = \{c_1, c_2, ..., c_n\}$;

$R^C \subseteq C \times C$ — множина внутрішніх відношень;

$R^i \subseteq C_\Theta \times C$ — множина зовнішніх вхідних відношень;

$R^O \subseteq C \times C_\Theta$ — множина внутрішніх вихідних відношень;

B — множина поведінки (або функцій) $B = \{b_1, b_2, ..., b_p\}$;

Ω — множина структур на компонентах, умови відношень, рамки (область дії) поведінки, $\Omega = \{w_1, w_2, ..., w_q\}$;

Θ — зовнішнє середовище з непорожньою множиною компонентів C_Θ, $C_\Theta \cap C = \varnothing$.

Ураховуючи введені позначення, можна переписати рівність (1) у вигляді:

$$ES = ES\left(\cup_{i=1}^{5} C_i, \cup_{i=1}^{5} R_i^C \cup_i \Delta R_j^C, \cup_{i=1}^{5} R_i^i, \cup_{i=1}^{5} R_i^O, \cup_{i=1}^{5} B_i \cup_j \Delta B_j, \cup_{i=1}^{5} \Omega_i, \cup_{i=1}^{5} \Theta_i\right), \qquad (2)$$

де $\cup$ — кон'юнкція множин; $\Delta R^C, \Delta B$ — нові внутрішні відношення та емергентна поведінка екосистеми, що виникають при кон'юнкції систем-складових; $j = C_n^2 = \frac{n!}{2!(n-2)!}$ — кількість систем, що входять у екосистему ПЗ.

2. Формально описати кожен елемент (множину) кортежу пропонується за допомогою таких теорій:

- для C, Θ — теорія множин та алгебричних 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) [13]. Необхідно врахувати, що поведінка програмної системи поділяється на статичну і динамічну.

3. Загальні рекомендації з формального опису систем-складових екосистеми ПЗ.

Система S_1 «*Програмне забезпечення*».

$S_1 = \left(C_1, R_1^C, R_1^i, R_1^O, B_1, \Omega_1, \Theta_1\right)$ — відкрита технічна абстрактна система.

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

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

Приклад опису множини C_1.

Нехай описуване ПЗ — це деяка система, призначена для збору інформації у виглядів відповідей експертів та/або користувачів з подальшою її обробкою. Тоді множина компонентів системи може мати такий вигляд:

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

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

c_{i1} = {files *.jsp, Web-browser, TomCat Web-Server}, $i = \overline{1,7}$;

c_{i2} ={tables}, $i = \overline{1,18}$;

c_{i3} ={libraries, executable component}, $i = \overline{1,16}$.

Загальна кількість компонентів системи — 41.

У цьому прикладі компонент є фізичною абстракцією, що матеріалізує клас (логічну абстракцію). Можливість більш докладного розгляду структури ПЗ з погляду рівнів інкапсуляції наведено в праці [8]. У кожному окремому випадку поняття «компонент програмного забезпечення» може відрізнятися залежно від потреб дослідників.

2. $R_1^c = \{\rightarrow\}$, де $\rightarrow$ — відношення базової залежності.

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

$$((c_{11} \rightarrow c_{nm}) = \Rightarrow (c_{11},\ c_{nm}) \hat{=}$$

$$\hat{=} (c_{nm} \xrightarrow{\varphi} c'_{nm}) \Rightarrow (c_{11} \xrightarrow{\psi} c'_{11})),$$

де φ, ψ — перетворення «зміна компонента».

Для того щоб розглядувана структура (кортеж) справді була системою, необхідно перевірити виконання властивостей рефлексивності та асиметричності для визначеного відношення.

В даному випадку рефлексивність відношення є очевидною. Властивість асиметричності також виконується, оскільки:

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

3. R_1^i = {інтеграція (повторно використовувані компоненти ПЗ, компоненти ПЗ); розробка та супровід (розробники, компоненти ПЗ); функціонування (апаратне забезпечення, компоненти ПЗ); створення нормативної бази (органи стандартизації, компоненти ПЗ); захист прав на інтелектуальну власність (держава, компоненти ПЗ)}.

4. R_1^O = {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)}.

5. Θ_1= {апаратне забезпечення, природні об'єкти та явища, соціальна система, економічна система, інші екосистеми ПЗ}.

6. B_1 = {множина поведінок (функцій) системи}.

7. Ω — множина структур на компонентах, умови відношень, рамки (область дії) поведінки, $\Omega = \{\omega_1, \omega_2, ..., \omega_q\}$ — описується залежно від контексту (для конкретної екосистеми).

Як уже зазначалось, поведінка програмної системи поділяється на статичну й динамічну та зручно описується за допомогою алгебри процесів реального часу [13].

Система S_1 «Програмне забезпечення»

$S_1 = \{C_1, R_1^c, R_1^i, R_1^0, B_1, \Omega_1, \theta_1\}$ — відкрита технічна абстрактна система.

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

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

Система S_2 «Апаратне забезпечення»

Аналогічно до системи «Програмне забезпечення» описується як

$S_2 = \{C_2, R_2^c, R_2^i, R_2^0, B_2, \Omega_2, \theta_2\}$ — технічна матеріальна система.

Компоненти системи та її поведінка залежать від конкретного розглядуваного апаратного забезпечення.

Економічна система S_4

$S_2 = \{C_4, R_4^c, R_4^i, R_4^0, B_4, \Omega_4, \theta_4\}$ — відкрита економічна система.

C_4 = {виробники ПЗ, продавці ПЗ, посередники, користувачі ПЗ, держава}.

Під час визначення множини внутрішніх відношень R_4^C необхідно враховувати макро- та мікроекономічні аспекти. Розглядаючи екосистему ПЗ як систему систем, будемо говорити про економічні відносини як певні зв'язки, в які незалежно від волі та свідомості вступають люди в процесі виробництва, розподілу, обміну та споживання (використання) програмного забезпечення. На макроекономічному рівні говорять про соціально-економічні, організаційно-економічні та виробничі відносини.

Отже, множину R_4^C можна подати як сукупність таких відношень:

R_4^C = {відносини власності на засоби виробництва; кооперація праці; спеціалізація праці; суспільне відтворення; договірні відносини; майнові відносини}.

Вхідні та вихідні відношення R_4^i, R_4^O визначаються екосистемою конкретного ПЗ. Серед процесів, які характеризують поведінку економічної системи на мікроекономічному рівні, можна назвати:

- аналіз прибутковості;
- розмір проекту;
- ідеальну завантаженість проекту;
- оптимальні трудові ресурси;
- аналіз вартості проекту;
- очікуване робоче навантаження.

Також визначають такі важливі показники, як строк окупності проекту, норма прибутку тощо.

Соціальна система S_5

$S_5 = \{C_5, R_5^c, R_5^i, R_5^0, B_5, \Omega_5, \theta_5\}$ — відкрита соціальна система.

C_5 = {індивіди (розробники, менеджери проекту), групи, суспільні організації}.

$R_5^C = \{R(p), SR(p)\}$

$R(p) = r : p \to P, \quad p \in P$ — відношення між індивідами p типу «один до одного», «один до багатьох», «багато до багатьох»; P — множина всіх індивідів.

$SR(p) = f : p \to F$, $p \in P$ — соціальна роль — відношення між особистістю p та множиною соціальних функцій F.

Вхідні та вихідні відношення R_5^i, R_5^O визначаються екосистемою конкретного ПЗ.

Поведінка B_5 соціальної системи може бути змодельована за допомогою множини взаємодіючих функцій. Соціальна функція Ф є множиною завдань та \ або дій у суспільстві, які можуть

виконуватись індивідами. Високорівневі соціальні функції можуть поділятися на дві категорії: суспільні та особисті. Робоча функція особистості в організації-виробнику ПЗ належить до множини суспільних. Слід зазначити великий вплив відчуттів (емоцій, мотивацій, ставлення) на поведінку індивідів. У праці [13] наведено модель поведінки, керованої мотивацією \ ставленням. Також необхідно враховувати особливості колективної поведінки (узгодженість, координація, групове мислення тощо).

Висновки

Розгляд екосистем програмного забезпечення з погляду системи систем є доцільним і дає можливість їх досліджувати на новому рівні — не просто як суму властивостей систем, що складають соціально-технічне середовище програмного забезпечення, а як системи, в якій можуть виникати нові, непередбачувані та корисні властивості. Нині даний момент в теорії системи систем є необхідним розроблення принципів і методів, які б дозволили досліджувати розвиток, використання та еволюцію системи систем. Так само питання еволюції є важливим і в екосистемах програмного забезпечення. Формалізований опис систем-складових екосистеми програмного забезпечення сприяє більш наочному представленню даних, яке, у свою чергу, спрощує аналіз функціонування та розвитку конкретних екосистем.

ЛІТЕРАТУРА

1. *Сидоров Н. А*. Экология программного обеспечения / Н. А. Сидоров // Інженерія програмного забезпечення. — К., 2010. — № 2. — С. 1–9.

2. *Сидоров Н.* Системная инженерия программного обеспечения / Н. Сидоров, М. Луцкий, И. Гученко // Інженерія програмного забезпечення. — 2010. — № 4. — С. 13–25.

3. *Grinenko O*. Models of software ecosystems / O. Grinenko // The Fifth World Congress “Aviation in the XXI century”. — Kiev, 2012. — P. 1.10.23–1.101.27.

4. *Grinenko O*. Software Ecosystems / O. Grinenko // Journal of the National Transport University. — Kiev: National Transport University. — 2012. — Vol. 26. — P. 508–512.

5. *Sidorov* N. Software Ecosystem Modeling / N. Sidorov, O. Grinenko // Journal “Software Engineering”. — Kiev : National Aviation University. — 2013. — Vol. 2 (14). — P. 38–48.

6. *Messershmitt D. G*., Szyperski C. Software Ecosystems: Understanding an Indispensable Technology and Industry. — London : MIT press, 2003. — 233 p.

7. Webber L., Wallance M. Green Tech: thow two and to plan implement sustain — able IT solutions. — FBACOM. — 2009. — 29 p.

8. *Velye Y*., Velye A., Elsenpeter R. Green IT: Rednee your Information systems Environmental input while adding to the bottom line. — 2008.

9. *Schulz G*. The Green and Virtual Data Center // Taylors Francis G. — 2009. — 218 p.

10. *Jansen S*, Bosch J. Understanding Software Ecosystems: A Strategic Modeling Approach // Proceedings of the Workshop on Software Ecosystems 2011. — P. 65–76.

11. *Jansen S*., Finkelstein A., Brinkkemper. A sense of community: A research agenda for software ecosystems // 31th International Conference on Software Engineering, New and Emerging Research Track. —2009. — P. 12–16.

12. *Duinkerken W*. Transaction Cost Economics in Software Ecosystems: some empirical evidence. — April 20, 2009. — P. 22–37.

13. *Wang Yingxu*. Software engineering foundations: a software science perspective. — Auerbach Publications, 2008. — 1420 p.

Стаття надійшла до редакції 29.08.2014.