

УДК 004.55

Д. С. НЕГУРИЦЯ

*Харківський національний університет радіоелектроніки, Україна*

## АДАПТИВНІСТЬ ТА АДАПТОВАНІСТЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМ КРИТИЧНОГО ПРИЗНАЧЕННЯ

*У статті визначено складові життєздатності систем критичного призначення, проаналізовано відомі методи побудови та забезпечення якості таких систем, наведені найбільш поширені визначення понять «адаптивність» і «адаптованість». Показано, що сучасні стандарти програмної інженерії трактують ці поняття в невиправдано вузькому формулюванні. Архітектура адаптивного програмного забезпечення систем критичного призначення побудована відповідно до зробленої в роботі класифікації механізмів адаптивності. Розроблено модель процесів взаємодії між користувачем та адаптивною системою.*

**Ключові слова:** адаптивність, адаптованість, модель користувача, інтерфейс користувача, якість програмного забезпечення, система критичного призначення, еволюція.

### Вступ

«У програмному забезпеченні все змінюється. Вимоги змінюються. Проекти змінюються. Технології змінюються. Склади команд змінюються. Члени команд змінюються. Проблеми не мають змоги змінитися самі по собі, бо такі зміни – скоріше є свідчення нездатності проблеми впоратися зі змінами зовнішнього світу» [1].

Великі програмні системи будуть продовжувати зростати та змінюватися, при цьому їх якість буде знижуватися, якщо вони перестануть реагувати на змінення в оперативній обстановці [2]. Альтернативою еволюції системи є її повна заміна, яка часто є не можлива. Вартість години простоювання великої системи досягає \$ 180.000 USD для Amazon, або майже \$ 6.500.000 доларів США для програмного забезпечення підтримки брокерських операцій [3].

Критичні об'єкти характеризуються тим, що розміри збитків або інших наслідків, які можуть виникнути в результаті порушення їх працездатності, збоїв і відмов у роботі, є неприйнятними для суспільства [4]. До таких об'єктів можна віднести військові об'єкти, екологічно небезпечні промислові виробництва, атомні станції, об'єкти транспорту, зв'язку, фінансово-кредитної сфери тощо.

Однією з особливостей сучасного етапу розвитку техніки є розповсюдження систем критичного застосування, збоїв та відмови яких несуть потенційну загрозу природі та людському суспільству. При цьому ключова роль у забезпеченні безпеки належить інформаційно-керуючим системам, які виконують функції запобігання, захисту та ліквідації наслідків аварійних ситуацій [5].

### 1. Еволюціонуючі системи критичного призначення

Згідно «законів» Лемана [6, 7] працююче ПЗ знаходиться в одній з двох стадій: або зазнає постійних змін, або деградує. В роботі [32] зазначається, що в процесі змін менше 20% часу витрачається на виправлення помилок, непомічених при розробці та тестуванні. Більше 80% тимчасових витрат пов'язано зі зміною системи у зв'язку або з змінами процесів реального світу, або з розширенням функціональності.

Процес супроводу системи критичного призначення найбільш тривалий і дорогий в її життєвому циклі [8, 9]. Стандарт ISO/IEC 14764 [10] визначає 4 категорії супроводу.

1. Корируючий супровід – виправлення помилок, виявлених в процесі експлуатації.

2. Адаптивний супровід - зміни ПЗ в зв'язку із змінами в оточенні.

3. Удосконалюючий супровід - модифікація ПО з метою поліпшення продуктивності, інтерфейсу, а також розширення функціональності системи.

4. Профілактичний супровід - модифікація ПО для виявлення та виправлення прихованої несправності до того, як несправність стане явною, підвищення здатності до супроводу в частині масштабування, поліпшення модульності структури програм. Останні 3 типи супроводу відносять до еволюції ПО [11].

ІТ-фахівці повинні сконцентрувати зусилля на вдосконаленні методів, методологій та інструментальних засобів, необхідних для розроблення, реалізації та підтримки критичних програмних систем, зда-

тних до успішного еволюціонування. Важливість еволюціонування критичних систем (evolving critical systems, ECS) підкреслюється рядом масштабних змін в програмній інженерії. Найбільш суттєві зміни можна охарактеризувати наступним чином:

– повсюдність програмного забезпечення - програмне забезпечення все більшого обсягу встановлюється в побутових пристроях, і збої цього програмного забезпечення безпосередньо зачіпають життя звичайних людей;

– програмне забезпечення все частіше стає критичним, навіть якщо воно не є складним;

– участь людей - в міру впровадження програмного забезпечення в управляючі системи, в роботі яких беруть участь люди, все більш важливою проблемою стає взаємодія людини з програмним забезпеченням;

– складність – залежності між компонентами програмного забезпечення ускладнюються, і велика частина програмного забезпечення, використовуваного в застосуваннях реального світу, взаємодіє з програмним забезпеченням сторонніх виробників;

– зростаюча швидкість еволюції визначається зростанням рівня вимог з боку користувачів; ринок програмного забезпечення безжальний до постачальникам, які не здатні дешево і швидко змінювати своє програмне забезпечення.

## 2. Адаптивність і адаптованість як складові життєздатності систем критичного призначення

Центральним поняттям теорії життєздатності систем є «життєздатність», близьке значення мають «життєстійкість», «життєзабезпечення», «життєвість» та ін. [12]. У кібернетичної моделі С. Біра [13] життєздатність визначається можливістю системи самоорганізовуватися, тобто змінюватися в потрібному темпі. Іноді під життєздатністю розуміють надійність і стійкість [14], або робастність програмного забезпечення, як стійкість системи до деяких змін навколишнього середовища [15].

Життєздатною називають таку систему, яка володіє рисами й адаптованості та адаптивності [16]. Адапована система пристосована до змін людиною в процесі життєвого циклу відповідно до вимог зовнішнього середовища [17]. Адаптивні системи - це такі системи, які виконують зміни самі. Побудова життєздатних систем дозволяє вирішувати проблеми складності програмного забезпечення.

Згідно Ч. Херрінга [18] життєздатна система, щоб справлятися зі складністю зовнішнього оточення, повинна бути адаптивною, пройшовши стадію адаптованості. Система за R. Orpermann [19] є адаптивною, «якщо вона здатна автоматично змінити

свої особливості відповідно до потреб користувача». Адаптивні системи контролюють взаємодію з користувачем і змінюють уявлення інтерфейсу або поведінку. За А. Jameson [20]: «адаптивна до користувача система - це інтерактивна система, яка адаптує свою поведінку до кожного користувача на основі нетривіальних висновків з інформації про цього користувача». Поняття «нетривіальні висновки» включає прямі пристосування (наприклад, зміни кольору шрифту по команді користувача). Ключовим для цієї роботи положенням є наслідок з А. Jameson - адаптивна програмна система повинна змінювати свою поведінку на основі інформації, отриманої в процесі спостереження за користувачем [21].

Міжнародні та вітчизняні стандарти, що регламентують вимоги до якості ПЗ (ISO / IEC 9126, серії ECSS -Q, COU- Н НКАУ 0012:2006 Вимоги до якості програмного забезпечення програмно-технічних комплексів критичного призначення) визначають адаптованість, як пристосованість програм та інформації баз даних до модифікації для експлуатації в різних апаратних та операційних середовищах без застосування інших дій або засобів, ніж ті, що призначені для цієї мети при первинній розробці в вихідній версії програмної системи.

Адапованість є однією з субхарактеристик для двох показників зовнішньої якості програмного забезпечення (ПЗ).

1. Супроводжуваність (Maintainability) – здатність ПЗ до модифікації, що може містити у собі виправлення, поліпшення або адаптацію ПЗ до змін середовища, вимог або функціональних специфікацій.

2. Мобільність (Portability), під якою розуміють:

– здатність ПЗ бути перенесеним з одного організаційного, апаратного або програмного середовища в інше із збереженням їх цілісності або з невеликими змінами функцій системи;

– процеси повторного використання готових програмних компонентів і засобів, а також інформації баз даних можливо в межах однієї архітектури апаратної і операційного середовища для розширення і зміни функцій системи і програмного продукту.

Існує велика кількість класифікацій адаптивних систем керування [22]. За критерієм адаптації можна розрізнити системи з еталонною моделлю і системи з екстремальної самонастроюванням. У перших з них регулятор адаптується таким чином, щоб замкнута система управління мала властивості, як можна більш близькі до заданих властивостей еталонної моделі. У других регулятор адаптується з метою отримати найкращі в тому чи іншому сенсі показники якості управління.

Класифікувати механізми адаптивності ПЗ слід

за кількома критеріями.

1. Внутрішня або зовнішня по відношенню до системи адаптивність [23]. Внутрішня – це можливість програмної системи використовувати типові механізми, – наприклад обробку винятків, для запуску певної програми, яка коригує поведінку системи. Зовнішня адаптивність досягається шляхом моніторингу різних атрибутів системи, таких як використання ресурсів, надійності та/або якості обслуговування, при цьому зовнішній по відношенню до системи механізм приймає рішення про внесення зміни.

2. Об'єкт змін у процесі адаптації:

– адаптивність поведінки, коли елементи системи можуть бути змінені або замінені, а структура системи при цьому зберігається [24];

– адаптивність структури, коли структура системи змінюється, а поведінка її елементів залишається тим же [25];

– ресурсна адаптація, коли поведінка системи змінюється за рахунок управління ресурсами з урахуванням потреб користувачів і несправностей [26].

3. Ступінь самоврядування характеризує можливість програмної системи зберігати ефективність без втручання користувача [27]:

– самоадаптація - можливість змінювати власну поведінку у відповідь на зміни у своєму операційному середовищі з метою забезпечення і підвищення функціональних можливостей [28];

– самоорганізація - здатність компонентів автономно взаємодіяти з іншими компонентами, щоб відповідати системним обмеженням [29];

– самоконфігурування – автоматична адаптація до динамічних змін шляхом вставки, заміни та видалення своїх компонентів [30];

– самозахист – здатність шляхом спостереження за зовнішнім середовищем помічати і реагувати на можливі атаки на систему;

– самовідновлення – автоматичне реагування і корегування у відповідь на програмні та апаратні збої [31];

– самооптимізація – можливість автономно оптимізувати системні ресурси в процесі моніторингу [31].

### 3. Архітектура адаптивного програмного забезпечення систем критичного призначення

Архітектура адаптивного програмного забезпечення призначена для підтримки розробників у виборі відповідних методів подання проектних рішень. Така архітектура складається з трьох основних компонентів: моделі користувача, моделі предметної області, а також моделі їх взаємодії (рис. 1).

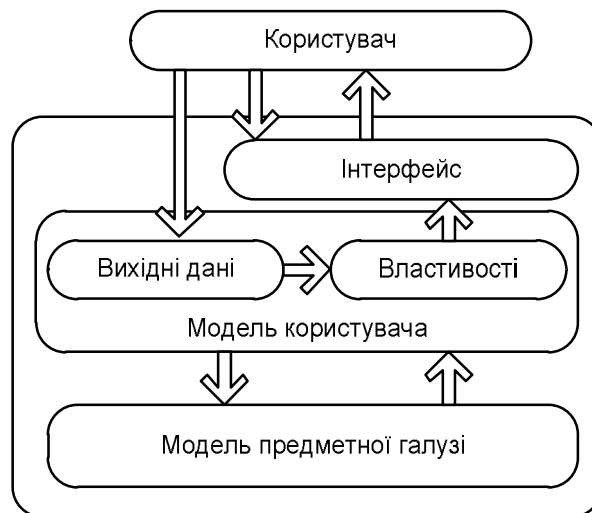


Рис. 1. Взаємодія компонентів адаптивного програмного забезпечення

Модель користувача представляє знання системи про користувача. Вона складається з трьох взаємопов'язаних компонентів: по-перше, модель знань, яка містить припущення системи щодо знань користувача про предметну галузь (домену). Наприклад, система може припускати, що користувач знає, як відкрити текстовий файл. Таким чином, ця інформація залежить від програми та домену. Другий компонент цієї моделі є модель профілю, який містить інформацію про вподобання користувача, інтереси та загальні знання. Третій компонент – психологічна модель, характеризує доменно-незалежні пізнавальні та афективні риси користувача.

Модель предметної області визначає аспекти системи, які є важливими для висновків, наприклад, функцій, які можуть бути змінені. Ці аспекти можуть бути описані на різних рівнях, таких як рівень завдань, логічному рівні, або фізичному рівні. Таким чином, модель предметної області є основою для всіх висновків та адаптації.

Модель взаємодії обробляє діалог між користувачем та програмою. Вона може фіксувати попередню взаємодію і містити механізми для виведення властивостей користувачів, механізми адаптації системи до них властивостей користувача, і механізми для оцінки адаптації.

Таким чином, ця модель описує різні види інформації, які необхідні для адаптації. Тим не менш, в деяких системах основні компоненти можуть бути не настільки помітно відокремлені одно від одного.

## Висновки

У зв'язку з вимогами постійних змін і з урахуванням складності систем критичного застосування виникає завдання вдосконалення методів, методологій та інструментальних засобів, необхідних для розробки, реалізації та підтримки критичних програмних систем. Такі системи повинні бути одночасно і адаптованими, тобто пристосованими до змін людиною в процесі життєвого циклу відповідно до вимог зовнішнього середовища, і адаптивними, тобто здатними пристосувати свою поведінку до кожного користувача на основі нетривіальних висновків з інформації про цього користувача. Питання побудови адаптивних систем, оцінки їх якості в існуючих стандартах інженерії програмного забезпечення викладені вельми у вузькій постановці. Адаптивність в стандартах розглядається в контексті пристосованості до зміни апаратно-програмного оточення, але не користувача, що в черговий раз підкреслює актуальність робіт по даному напрямку.

## Література

1. Beck, K. *Extreme Programming Explained: Embrace Change* [Text] / K. Beck. // Addison-Wesley. – 1999. – P. 196.
2. Lehman, M. M. *Rules and tools for software evolution planning and management* [Text] / M. M. Lehman, J. F. Ramil // *Ann. Softw. Eng.* – 2001. – P. 15–44.
3. Hinchey, M. *Evolving critical systems. Technical Report* [Text] / M. Hinchey, L. Coyle // *Lero-TR-2009-00, Lero.* – 2009. – 120 p.
4. Харченко, В. С. *Нормирование и оценка безопасности информационных и управляющих АЭС: регулирующие требования к программному обеспечению* [Текст] / В. С. Харченко, М. А. Ястребинецкий, В. Н. Васильченко // *Ядерная и радиационная безопасность.* – 2002. – № 1. – С. 18–33.
5. *Доказательная независимая верификация и оценка скрытых дефектов критического программного обеспечения на основе диверсифицированного измерения инвариантов* [Текст] / Б. М. Конорев, В. В. Сергиенко, Г. Н. Чертков, Ю. Г. Алексеев // *Радиоэлектронные и компьютерные системы.* – 2009. – № 7. – С. 192–199.
6. Леман, М. М. *Программы, жизненные циклы и законы эволюции программного обеспечения* [Текст] : пер. с англ. / М. М. Леман. – ТИИЭР, Техника программного обеспечения. – М. : Мир, 1980. – Т. 68, № 9. – С. 26–45.
7. *Metrics and laws of software evolution—the nineties view* [Text] / M. M. Lehman, J. F. Ramil, P. D. Wernick, D. E. Perry, and W. M. Turski // *Proc. 4th International Software Metrics Symposium (METRICS '1997).* – P. 20–32.
8. Pigosky, T. M. *Practical Software Maintenance.* [Text] / T. M. Pigosky. – New York : John Wiley & Sons, 1996. – 184 p.
9. *Guide to Software Engineering Base of Knowledge (SWEBOK) [Text].* – IEEE Computer Society, 2004. – 200 p.
10. Mookerjee, R. *Maintaining enterprise software applications* [Text] / *Communications of the ACM.* – Nov 2005. – Vol. 48, Issue 11. – P. 75-79.
11. *Software maintenance as part of the software life cycle/Tufts University.-2003 [Электронный ресурс]* / K. Erdil, E. Finn, K. Keating, J. Meattle, S. Park, D. Yoon. – Режим доступа: [http://hepguru.com/maintenance/Final\\_121603\\_v6.pdf](http://hepguru.com/maintenance/Final_121603_v6.pdf)
12. Разумовский, О. С. *Проблемы жизнеспособности систем [Электронный ресурс]* / О. С. Разумовский, М. Ю. Хазов. – Режим доступа: [http://www.philosophy.nsc.ru/journals/humscience/1\\_98/01\\_RAZUM.htm](http://www.philosophy.nsc.ru/journals/humscience/1_98/01_RAZUM.htm).
13. Beer, S. *Brain of the Firm* [Text] / S. Beer. – Wiley, Chichester. – 1981. – 432 c.
14. Лачинов, В. М. *Информодинамика [Электронный ресурс]* / В. М. Лачинов, А. О. Поляков. – Режим доступа: <http://www.inftech.webservis.ru/it/information/informodynamics>.
15. Izurieta, C. *A multiple case study of design pattern decay, grime, and rot in evolving software systems* [Text] / C. Izurieta, J. M. Bieman // *Software Quality.* – 2013. – Vol. 21, Issue 2. – P. 289-323.
16. Черняк, Л. *Адаптируемость и адаптивность [Текст]* / Л. Черняк // *Открытые системы.* – 2004. – № 9. – С. 30-35.
17. De Bra, P. *Adaptive Hypermedia: From Systems to Framework* [Text] / P. De Bra, P. Brusilovsky, G.-J. Houben // *ACM Computing Surveys.* – 1999. – Vol. 31, № 4. – P. 12.
18. Херринг, Ч. *Жизнеспособные программы [Текст]* / Ч. Херринг, Л. Черняк // *Открытые системы.* – 2004. – № 9. – С. 77-80.
19. Oppermann, R. *Adaptively supported adaptability* [Text] / R. Oppermann // *International Journal of Human Computer Studies.* – 1994. – № 40(3). – P. 455–472.
20. Jameson, A. *Systems That Adapt to Their Users: An Integrative Overview* [Text] / A. Jameson // *International Conference on User Modeling, Johnstown, USA, 22-26 June, 2003.* – Saarland University, 2003. – 253 p.
21. *Flexible Content Adaptation System Using a Rule-Based Approach* [Text] / J. He [et al] // *IEEE Transactions on Knowledge and Data Engineering.* – 2007. – Vol. 19, № 1. – P. 127-140.
22. Усков, А. А. *Интеллектуальные системы управления на основе методов нечеткой логики.* [Текст] / А. А. Усков, В. В. Круглов. – Смоленская городская типография, 2003. – 177 с.
23. Tarvainen, P. *Open Access Adaptability Evaluation at Software Architecture Level* [Text] / P. Tarvainen // *The Open Software Engineering Journal.* – 2008. – № 2. – P. 1-30.
24. Masciadri, L. *Frameworks for the Development of Adaptive Systems: Evaluation of Their Adapta-*

bility Feature Software Metrics [Text] / L. Masciadri, C. Raibulet // *Proceedings of the 4th International Conference on Software Engineering Advances, Porto, Portugal, 20-25 Sept. 2009.* – P. 41-50.

25. Roman, M. *Reflective Middleware: From Your Desk to Your Hand* [Text] / M. Román, F. Kon, R. H. Campbell // *IEEE Distributed Systems Online (Special Issue on Reflective Middleware).* – 2001. – Vol. 2, № 5. – P. 1-13.

26. Aksit, M. *Dynamic, adaptive and reconfigurable systems overview and prospective vision* [Text] / M. Aksit, Z. Choukair // *Proceedings of the 23rd International Conf. on Distributed Computing Systems Workshops, Providence, RI, USA, 19-22 May 2003.* – P. 84-89.

27. Zhang, Y. *Software architecture design of an autonomic system* [Text] / Y. Zhang, A. Liu, W. Qu // *Proceedings of the 5th Australasian Workshop on Software and System Architectures, Melbourne, Australia, 13-14 April 2004.* – P. 5-11.

28. Andrade, L. *An architectural approach to autoadaptive systems* [Text] / L. Andrade, J. L. Fiadeiro // *Proceedings of the 22nd International Conference on Distributed Computing Systems, Vienna, Austria, 2-5 July 2002.* – P. 439-444.

29. Herrmann, K. *Meshmd1-a middleware for self-organization in ad hoc networks* [Text] / K. Herrmann // *Proceedings of the 23rd International Conf. on Distributed Computing Systems Workshops, Providence, RI, USA, 19-22 May 2003.* – P. 446-451.

30. *Realizing multi-dimensional software adaptation* [Text] / P. K. McKinley, E. P. Kasten, S. M. Sadjadi, Z. Zhou // *Proceedings of the ACM Workshop on Self-Healing, Adaptive and Self-MANaged Systems, Held in Conjunction with the 16th Annual ACM International Conference on Supercomputing, New York City, 22-26 June 2002.* – P. 8.

31. *Containment units: A hierarchically composable architecture for adaptive systems* [Text] / J. M. Cobleigh, B. S. Lerner, L. J. Osterwell, A. Wise // *Proceedings of the ACM SIGSOFT Symposium on the Foundations of Software Engineering, Charleston, South Carolina, USA, 18-22 November 2002.* – P. 159-165.

32. De Bra, P. *Adaptive Hypermedia: From Systems to Framework* [Text] / P. De Bra, P. Brusilovsky, G.-J. Houben // *ACM Computing Surveys.* – 1999. – Vol. 31, № 4. – P. 12.

*Поступила в редакцію 28.02.2013, рассмотрена на редколлегии 25.03.2013*

**Рецензент:** д-р техн. наук, проф., зав. каф. економіки і маркетингу В. М. Вартачан, Національний аерокосмічний університет ім. Н.Е.Жуковського «ХАІ», Харків, Україна.

## АДАПТИВНОСТЬ И АДАПТИРУЕМОСТЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СИСТЕМ КРИТИЧЕСКОГО НАЗНАЧЕНИЯ

*Д. С. Негурица*

В статье определены составляющие жизнеспособности систем критического назначения, проанализированы известные методы построения и обеспечения качества таких систем, приведены наиболее распространенные определения понятий «адаптивность» и «адаптируемость». Показано, что современные стандарты программной инженерии трактуют эти понятия в неоправданно узкой формулировке. Архитектура адаптивного программного обеспечения систем критического назначения построена в соответствии с произведенной в работе классификацией механизмов адаптивности. Разработана модель процессов взаимодействия между пользователем и адаптивной системой.

**Ключевые слова:** адаптивность, адаптируемость, модель пользователя, интерфейс, качество программного обеспечения, система критического назначения, сопровождение программного обеспечения, эволюция.

## ADAPTABILITY AND ADAPTABILITY OF SOFTWARE CRITICAL SYSTEMS

*D. S. Nehurytsia*

This article outlines the components of the viability of critical supplies, analyzed vi-house construction methods and quality assurance of these systems are the most common definition of "adaptability" and "adaptability". It is shown that the current standards of software engineering interpret these concepts in unreasonably narrow definition. Architecture of adaptive software critical systems provides a critical purpose built according to the classification made in the mechanisms of adaptability. A processes model of interaction between a user and the adaptive system has been engineered.

**Key words:** adaptive, adaptability, quality software system, critical applications, maintenance, evolution, user model, user interface.

**Негурица Дмитро Сергійович** – аспірант кафедри програмної інженерії Харківського національного університету радіоелектроніки, Україна, e-mail: neguritsa@gmail.com.