

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ, СИСТЕМНИЙ АНАЛІЗ ТА КЕРУВАННЯ

УДК 004.415.2

О.А. Авраменко

ЗАСТОСУВАННЯ МОДЕЛЕ-КЕРОВАНОГО ПІДХОДУ В РЕДОКУМЕНТУВАННІ ПРО- ГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Вступ

Процеси, зворотні до створення програмного забезпечення (ПЗ) – реінженерія, реструктуризація, редокументування, базуються на дослідженні вже існуючих програмних продуктів. Істотне поширення зворотні процеси отримали у 80-х роках минулого століття у зв'язку зі збільшенням кількості супроводжуваного та успадкованого ПЗ, що піддається модифікації, переробці або повторному використанню [1–4].

Результат реінженерії значною мірою залежить від повноти та якості документації успадкованого ПЗ [5]. На практиці таке ПЗ, найчастіше, не має документації або наявна документація є неповною і застарілою. Тому задачею редокументування стає забезпечення розробників, залучених до проектів реінженерії, достатньою і адекватною документацією успадкованого ПЗ, створеною на основі відновленої у зворотній інженерії інформації.

Реінженерія ПЗ включає процеси зворотної інженерії, які забезпечують відновлення інформації з успадкованого ПЗ, та прямої інженерії, яка дає можливість створення нового ПЗ і, зазвичай, є процесами технології розробки ПЗ. Саме технологія прямої інженерії визначає вимоги до складу, структури та змісту документів. Тому реалізація редокументування полягає в створенні документації, яка відповідає вимогам технології, вибраної для прямої інженерії [6].

Постановка задачі

Метою статті є дослідження широко відомого в програмній інженерії моделі-керованого підходу при застосуванні його для редокументування, орієнтованого на технологію розробки програмного забезпечення.

Документація технологій розробки ПЗ

Розглянемо вимоги сучасних технологій розробки RUP, MSF, CDM Oracle та гнучких технологій [7–9] до складу документації, структури та змісту окремих документів.

Документи технологій розробки складаються з документів фаз і належать до однієї з трьох категорій: розробки, продукту і процесу [10]. Склад документації регламентується переліком документів для кожної з категорій. Документ у переліку характеризується назвою, стислим описом та зазначенням фази, до якої він належить.

Технології RUP [7] і MSF [8] використовують перелік документації розробки та процесу з детальним описом змісту кожного з документів, а документація продукту розглядається як складова комплексу поставки ПЗ користувачу. Технологія CDM [17] використовує повний перелік документації розробки і продукту також з детальним описом змісту кожного з документів, а документацію процесу не регламентує. У гнучких технологіях склад документації – це список рекомендованих документів, який може застосовуватися повністю або частково залежно від вимог замовника та умов проекту (обсяг, складність, вартість, очікуваний час існування проекту, характер взаємодії з іншими командами розробників). У гнучких технологіях керування процесом розробки здійснюється завдяки особистому спілкуванню учасників, тому документація процесу не використовується.

Структура і склад документів технологій подаються в таких формах: загальна характеристика документа, перелік пунктів змісту з їх описом, шаблон та зразок документа. У технологіях розробки використовується одна з наведених форм або їх комбінації (таблиця). Загальна характеристика документа – це стислий неструктурований опис змісту документа, який зазначає ті аспекти, які має містити документ. Перелік пунктів змісту документа – це формалізований список пунктів, які повинні розкриватися в документі. Кожен пункт може мати характеристику, яка дає змогу уточнити його зміст. Шаблон – це формалізований опис

Таблиця. Форми структури і складу документів

Форма подання структури і змісту документа	Технологія розробки			
	RUP	MSF	CDM	Гнучка
Загальна характеристика документа	+	–	+	+
Перелік пунктів змісту документа	–	–	+	–
Шаблон документа	+	+	–	–
Зразок документа	+	–	–	–

структури і змісту документа у вигляді наперед визначених структурних елементів та описів їх вмісту. Зразок документа — це результат правильного заповнення шаблону документа в одному з проектів. Він дає можливість створювати документи за аналогією.

Проведений аналіз документації сучасних технологій розроблення RUP, MSF, CDM встановлює властивості документації, які суттєві з точки зору редокументування і які притаманні:

- документам всіх технологій — належність документів до певних фаз технологій, склад ідентифікаційних елементів (наприклад, номер версії та дата випуску). Ці властивості впливають із загальних вимог до документів ПЗ всіх технологій розробки;
- усім документам технології — склад службових структурних елементів (наприклад, глосарій і зміст), взаємне розміщення ідентифікаційних, службових та змістовних (наприклад, глави, параграфи, пункти) структурних елементів у документі, нотація документів. Ці властивості впливають із вимог, які висуває певна технологія розробки до всіх своїх документів;
- певному документу технології — структура документа, типи подань, які складають вміст документа. Ці властивості впливають із вимог певної технології розробки до певного документа.

Моделі документів у контексті моделі-керованого підходу

Моделі-керований підхід MDD (Model-Driven Development) застосовується в процесах прямої інженерії і полягає в поданні розробки ПЗ як перетворення узгоджених моделей, що описують ПЗ на різних рівнях абстракції [11]: незалежної від платформи моделі, орієнтованої на предметну область, платформено-залежної моделі і моделі реалізації (програмного коду). Такий підхід дає можливість повторно використовувати моделі верхніх рівнів, а при високій мірі формалізації та стандартизації моделей — автоматизувати їх перетворення.

Документ також можна подавати у вигляді моделі, яка формалізовано описує документ, враховуючи загальні вимоги до нього та вимоги, суттєві з точки зору редокументування. Ці вимоги повинні бути орієнтованими на технологію

розробки. Тоді створення документа можна вважати як послідовне уточнення його моделі.

Досліджуючи загальну концепцію документа ПЗ та його складові, а також виходячи з наведених рівнів властивостей документації технологій розроблення, для моделі-керованого редокументування пропонуємо використовувати сукупність моделей документа ПЗ, розміщених на різних рівнях, а саме:

- модель документа ПЗ M_G на рівні загальної концепції документа ПЗ, яка описує загальні аспекти змісту і структури документа ПЗ із врахуванням редокументування незалежно від вимог технологій розробки;
- метамодель документа технологій розробки M_T на рівні вимог технологій розроблення, яка інтерпретує модель M_G деталізацією вмісту і структури документа ПЗ із врахуванням загальних вимог технологій розробки R ;
- модель документа технології розробки M_D на рівні вимог до певного документа вибраної технології, яка є екземпляром метамоделі документа технологій розробки M_T . Ця модель дає можливість описувати вміст, структуру і форму подання конкретного документа технології як метадані (метаописи), які зумовлюють створення екземплярів документів при редокументуванні, у тому числі за допомогою засобів автоматизації. Модель M_T є метамоделлю документа ПЗ і описується засобами мови моделювання UML.

Модель документа програмного забезпечення

У галузевих стандартах документ ПЗ визначається як одиниця інформації, що унікально ідентифікується, розроблена для певного кола користувачів з певною метою і записана на будь-якому носії інформації [12].

Модель документа ПЗ M_G можна зобразити трійкою $M_G = \langle S, C, P \rangle$, де S — структура документа; C — вміст документа (інформаційне наповнення); P — форма подання документа (зовнішній вигляд).

Структура документа S — це упорядкована множина $S = \{s_i | i = 1, \dots, M\}$, де s_i — структурні елементи документації (СЕД), які можна розглядати як об'єднання $S = S_b \cup S_{id} \cup S_g$ трьох підмножин СЕД, які не перетинаються:

- основних (S_b), які визначаються інформаційним вмістом документа (наприклад, розділи, підрозділи та пункти документа);
- ідентифікаційних (S_{id}), які призначені для ідентифікації документа (наприклад, назву та номер версії документа);
- загальних (S_g), які необхідні для пошуку та навігації в документі (наприклад, глосарій та зміст документа).

Множина S утворює дерево G , яке складається з вузлів (СЕД) та ребер, що відображають відношення ієрархічної упорядкованості між СЕД:

$$G = \langle S, E \rangle,$$

де E – множина лінійно упорядкованих ребер дерева.

Основні та загальні СЕД можуть містити вкладені СЕД, порядок розміщення яких задається списком ребер. Наприклад, для кореневого СЕД-дерева s_d список ребер має вигляд

$$((s_d, s_1), (s_d, s_2), \dots, (s_d, s_i), \dots, (s_d, s_j), (s_d, s_n)),$$

де s_1, \dots, s_n – СЕД першого (верхнього) рівня (зміст, розділи, додатки, предметний вказівник і т. ін.).

Вміст документа C подано множиною $C = \{c_j \mid j = 1, \dots, L\}$, де c_j – інформаційний елемент документа (ІЕД) – логічно завершений інкапсульований елемент, який містить інформацію про ПЗ, наприклад діаграма класів, об'єктів, опис інтерфейсів. Між елементами множин C і S визначена відповідність O : якщо задано певний ІЕД $c \in C$, то визначено й СЕД $s \in S$. Відповідність O визначена для будь-якого елемента з множини C та є відображенням $O: C \rightarrow S$. ІЕД формується включенням до нього подання, яке є інформацією про частини ПЗ або про ПЗ у цілому. При редокументуванні подання береться з успадкованого (існуючого) ПЗ з використанням, у тому числі, засобів зворотної інженерії. Прикладами подань можуть бути перелік вимог, діаграми варіантів використання, компонентів та класів, опис класу, зображення користувацького інтерфейсу, схема бази даних (БД) і т. ін.

Подання v має значення z (текст, рисунок, таблиця, діаграма, анімація або звук) і характери-

зується властивостями p_1, \dots, p_n (тип, нотація, фізичне розміщення значення подання):

$$v = \langle z, (p_1, \dots, p_n) \rangle.$$

При редокументуванні ПЗ подання v_i з множини V можуть відповідати одному або кільком елементам c_j множини C . Відповідність множин V і C документа можна записати у вигляді матриці $A[K, L]$, де K – кількість елементів множини V ; L – кількість елементів множини C ; $A = \{a_{ij} \mid a_{ij} = \{0, 1\}, 1$ означає, що v_i відповідає c_j , 0 – в протилежному випадку}.

Крім подання, ІЕД може включати пояснення ex , яке має тільки текстове значення: $z(ex)$, наприклад, примітка, підписи таблиць і рисунків. Пояснення документа утворюють множину $EX = \{ex_i \mid i = 1, \dots, M\}$.

Таким чином, $\forall c_k \in C$ правильне співвідношення

$$c_k \Leftrightarrow \bigcup_{i=1}^L v_i^k \vee \bigcup_{j=0}^M ex_j^k,$$

де $c_k \in C$; $v_i^k \in V$ та відповідає k -му елементу множини C ; $ex_j^k \in EX$ та відповідає k -му елементу множини C .

Форма подання документа визначається правилами зовнішнього оформлення, які відповідають стандартизованим або корпоративним стилям оформлення та залежать від формату носія документа.

Метамодель документа технологій розробки програмного забезпечення

Метамодель документа технологій розробки ПЗ M_T створюється на основі розробленої моделі документа ПЗ M_G і вимог, спільних для

технологій розробки $R = \bigcap_{i=1}^N R_i$, де R_i – вимоги

i -ї технології. Метамодель $M_T = \langle C(R^c), S(R^s),$

$P(R^p) \rangle$ під час її перетворення параметризується за трьома складовими документа: вмістом, структурою, формою подання, на основі відповідних вимог R^c , R^s , R^p . Для вмісту параметром виступає нотація подань p_{notation} , для

структури – частина графа S , яка визначена для множин ідентифікаційних S_{id} і загальних S_g СЕД, $S'_1 = \langle S_{id} \cup S_g, E'_1 \rangle$, де $E'_1 \subset E$, для форми подання – множина правил форматування F , які встановлюють єдині правила оформлення для документів технологій. Таким чином, параметризована метамодель має вигляд $M_T(p_{notation}, S'_1, F)$.

До спільних вимог технологій розробки моделі M_T додаються властивості $prop = \langle p_{tech}, p_{phase}, p_{name} \rangle$, пов'язані з процесом розробки: назва технології p_{tech} , до складу якої входить документ, назва фази p_{phase} , на якій він створюється, і назва документа p_{name} .

Модель документа технології розробки програмного забезпечення

Модель документа технології M_D побудована на основі моделі M_T і вимог i -ї R_i технології розробки до документів. Модель $M_D = \langle C(R_i^c), S(R_i^s), P \rangle$ під час перетворення параметризується за двома складовими – вмістом і структурою – на основі вимог R_i^c і R_i^s . Для вмісту параметрами виступають множина

типів подань p_{type} , пояснень EX , для структури – частина графа S , яка визначена для основних СЕД S_b : $S'_2 = \langle S_b, E'_2 \rangle$ (де $E'_2 \subset E$), матриця $A[K, L]$ відповідності C і V і відображення $O: C \rightarrow S$.

Таким чином, із врахуванням параметризації метамоделі M_T модель M_D має вигляд $M_D(p_{notation}, p_{type}, A, EX, S, O, F)$.

Реалізація підходу моделі-керованої розробки в редокументуванні

Редокументування RD відповідно до моделі-керованого підходу розглядається як процес створення множини документів D_{RD} успадкованого ПЗ перетворенням моделей M документів з формуванням їх вмісту на основі множини подань V успадкованого ПЗ. Перетворення моделей керується двома параметрами – вибраною технологією розробки T і фазою Ph . Таким чином, маємо $D_{RD} = RD(M, V, T, Ph)$.

Реалізація редокументування, яка ґрунтується на моделі-керованому підході, полягає у виконанні таких послідовних перетворень (рис. 1):

- моделі документа M_G в метамодель M_T документа технологій розробки ПЗ – конкрети-

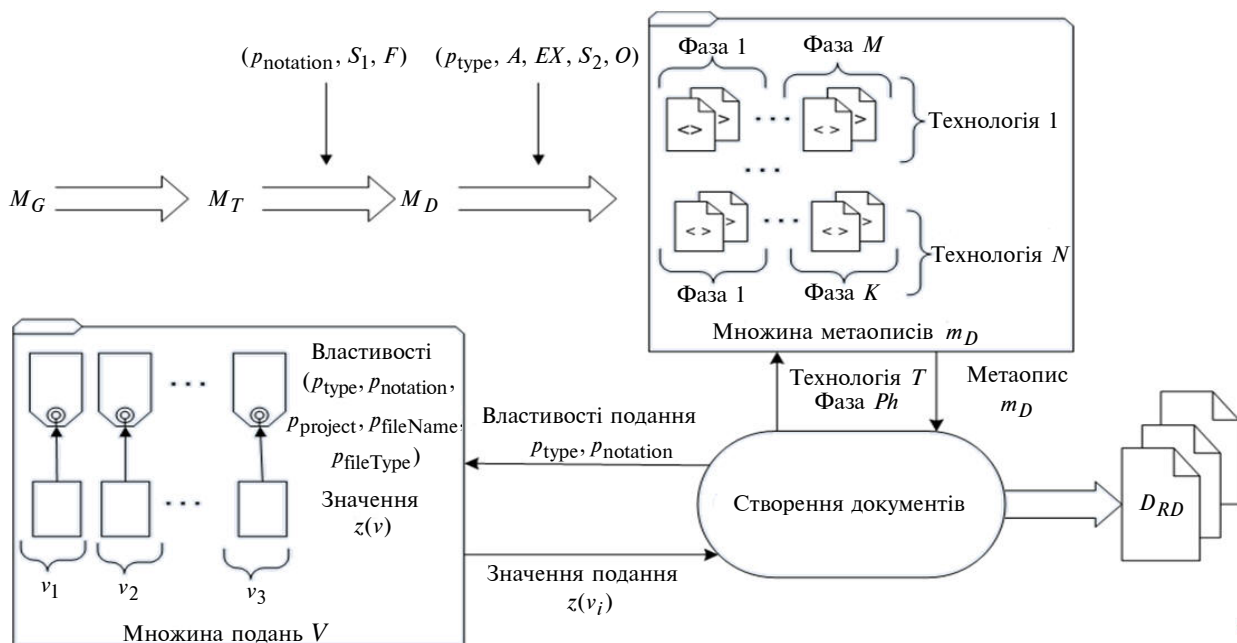


Рис. 1. Створення документів D_{RD} успадкованого ПЗ згідно з моделі-керованим підходом

зацією із врахуванням вимог, загальних для всіх технологій розроблення R ;

- метамоделі M_T в модель документів M_D технології конкретизацією із врахуванням вимог вибраної технології розробки R_i ;

- моделі M_D у метаописи m_D документів вибраної технології розробки підстановкою в параметри M_D фактичних значень, які визначаються вимогами R_{ij} до j -го документа.

Останній крок реалізації редокументування – створення документів D_{RD} успадкованого ПЗ як екземплярів метаописів m_D .

Перетворення моделі документа M_G в метамодель M_T документа технологій розробки ПЗ здійснюється відображенням елементів моделі M_G в елементи моделі M_T на мові UML та конкретизацією моделі M_G із врахуванням загальних вимог технологій розробки R . Якщо

загальні вимоги технологій до документів не змінюються, то метамодель M_T можна побудувати один раз, а потім застосовувати кожного разу при визначенні моделі M_D конкретної технології. Відображення дає можливість отримати M_G у вигляді діаграми класів на мові UML. Структура документа в моделі M_G , зображена у вигляді дерева, відображається в ієрархію класів метамоделі M_T із суперкласом “СЕД” (рис. 2).

При конкретизації моделі M_G на мові UML до неї додаються елементи, які описують параметри моделі M_T . Наприклад, параметр ($p_{notation}$) описується перелічувальним типом “Нотація подань” для переліку можливих значень параметра. Для опису параметра S'_1 до моделі впроваджуються класи “Назва проекту”, “Версія”, “Дата випуску”, “Автор”, щоб задати множину S_{id} ІЕД технологій розробки.

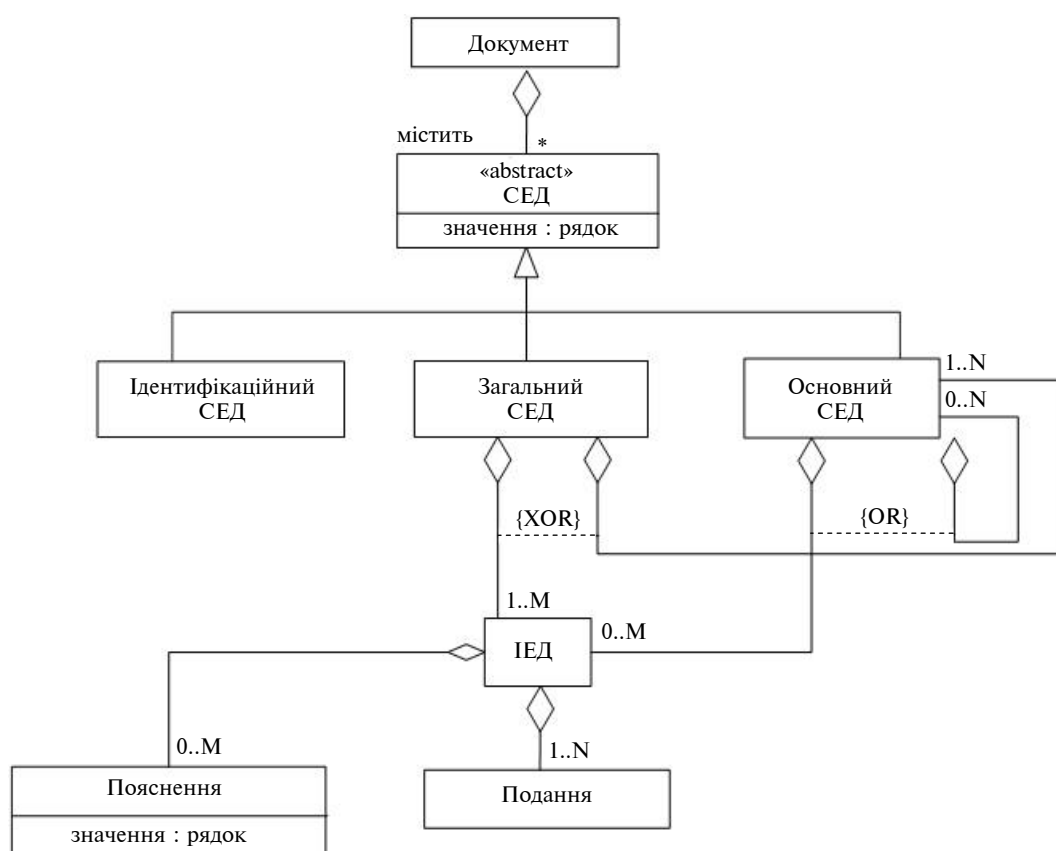


Рис. 2. Модель документа ПЗ M_G на мові UML

Перетворення метамоделі M_T в модель документа M_D технології здійснюється визначенням значень S_g , F , діапазону значень параметра $p_{notation}$ і діапазонів значень властивостей документа p_{tech} , p_{phase} , p_{name} . Отримана модель описує узагальнений документ конкретної технології розробки.

Розглянемо перетворення моделі документів M_D технології розробки RUP. Значення параметра S_g визначається множиною службових СЕД всіх документів технології. У технології RUP до елементів цієї множини належать “Зміст” і “Вступ”. При побудові моделі M_D до неї додаються класи “Зміст” і “Вступ”, які є підкласами абстрактного класу “Загальний СЕД”. Діапазон значень параметра $p_{notation}$ встановлюється значенням перелічуваного типу “Нотація подань” (пакет “Властивості подань” ($p_{notation}$)), який визначається вимогами технології RUP: UML, текст, таблиця. Діапазони значень властивостей документа p_{tech} , p_{phase} , p_{name} встановлюються значеннями перелічуваних типів “Назва технології”, “Назва фази” і “Назва документа” (пакет “Властивості документа” ($prop$)), які визначаються вимогами технології RUP.

Перетворення моделі M_D в метаопис m_D документів вибраної технології розробки R_j здійснюється визначенням для кожного з них значень параметрів моделі M_D . Засобом побудови метаописів є спеціально розроблена розширенням мови XML мова RDDDL (Redocumentation Document Description Language) [13]. Мова RDDDL містить засоби для опису СЕД, ІЕД, подань, пояснень та властивостей документа.

При створенні документа на основі властивостей подань, які містяться в метаописі до-

кумента, виконується пошук необхідних подань у множині V подань успадкованого ПЗ, які отримані засобами зворотної інженерії. Знайдені подання розміщуються в структурних елементах документа.

Для документів у межах однієї технології розробки перетворення моделей виконуються одноразово і документи створюються на основі готових метаописів з використанням для цього процесів трансляції та збирання, які можуть бути автоматизовані. Для автоматизації пропонується архітектура засобів [14], які належать до CARSE та виконують такі функції: підготовку метаописів документів для редокументування m_D , підготовку подань V успадкованого ПЗ та створення документів D_{RD} .

Висновки

Редокументування, орієнтоване на вимоги технологій розроблення, дає можливість здійснювати цілеспрямоване документальне забезпечення процесів прямої інженерії, що дає змогу уникнути зайвих витрат під час реінженерії, які зараз необхідні для створення відповідної документації.

Застосування моделі-керованого підходу для такого редокументування спрощує редокументування за рахунок повторного використання розроблених моделей. Так, для розробки моделей документів, орієнтованих на конкретну технологію, може використовуватися тільки одна метамодель, яка враховує загальні вимоги технологій розробки.

Можливе повторне використання моделі при проведенні редокументування в різних проектах реінженерії, але орієнтованих на одну технологію і фазу прямої інженерії.

При автоматизації процесів редокументування спрощується створення документів, які мають однакову структуру і зміст, але різну форму подання (формат), оскільки вони генеруються на основі однакових моделей.

Е.А. Авраменко

ПРИМЕНЕНИЕ МОДЕЛЕ-УПРАВЛЯЕМОГО ПОДХОДА В РЕДОКУМЕНТИРОВАНИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Рассматривается реализация редокументирования наследуемого программного обеспечения, которая

O.A. Avramenko

APPLYING MODEL-DRIVEN APPROACH IN SOFTWARE REDOCUMENTATION

This paper considers redocumentation of legacy software based on the model-driven approach. We

основана на моделі-управляемом подходе. Описаны модели документов разного уровня абстракции с учетом требований технологии разработки программного обеспечения. Предложены средства последовательного преобразования моделей, что позволяет получить метаописания документов, используемые средствами автоматической генерации документов.

describe various abstract levels of document models taking into account requirements of software development technology. In addition, sequential transformations of models allow obtaining metadescriptions of documents used by tools of documents automatic generation.

1. *Chikofsky E.J., Cross J.H.* Reverse Engineering and Design Recovery: A Taxonomy // IEEE Software. — 1990. — January. — P. 13–17.
2. *Sneed H.M.* Software Renewal: A Case Study // Ibid. — 1984. — 1, N 3. — P. 56–63.
3. *Tilley S.R., Smith D.B.* Perspectives on Legacy System Reengineering. — Pittsburgh, PA: Reengineering Center, SEI, Carnegie Mellon University, 1995. — 146 p.
4. *Сидоров Н.А.* Восстановление, повторное использование и переработка программного обеспечения. I // УСиМ. — 1998. — № 3. — С. 74–83.
5. *Сидоров Н.А., Хоменко В.А., Авраменко Е.А.* Реинженерия проектов программного обеспечения // Пробл. програмування. — 2006. — № 2-3. — С. 31–38.
6. *Сидоров Н.А., Авраменко Е.А.* Метод и средства редокументирования наследуемого программного обеспечения // Там же. — 2008. — № 2-3. — С. 229–238.
7. *Якобсон А., Буч Г., Рамбо Дж.* Унифицированный процесс разработки программного обеспечения / Пер. с англ. — СПб.: Питер, 2002. — 496 с.
8. *Microsoft Solutions Framework.* Модель процессов MSF. Версия 3.1. — Корпорация Майкрософт (Microsoft Corp.), 2002. — 44 с.
9. *Oracle Method®.* CDM Classic Method Handbook. Release 2.6.0. — Oracle corp., 2000. — 248 p.
10. *ГОСТ Р ИСО/МЭК ТО 9294–93.* Информационная технология: Руководство по управлению документированием программного обеспечения. — Изд. официальное. Госстандарт России. — М.: Изд-во стандартов, 1994. — 15 с.
11. *Mellor S.J., Clark A.N., Futagami T.* Model-Driven Development // IEEE Software. — 2003. — 20, N 5. — P. 14–18.
12. *ДСТУ 4302:2004 (ISO/IEC 6592:2000(MOD)).* Інформаційні технології. Настанови щодо документування комп'ютерних програм. — К.: Держспоживстандарт, 2005. — 30 с.
13. *Авраменко Е.А.* Средство редокументирования наследуемого программного обеспечения // Вісн. НТУУ "КПІ". Сер. Інформатика, управління та обчислювальна техніка: Зб. наук. праць. — К.: "ВЕК+", 2007. — № 47. — С. 100–107.
14. *Авраменко О.А.* Архітектура засобів редокументування успадкованого програмного забезпечення. — Вісн. НАУ. — 2007. — № 3-4 (33). — С. 58–62.

Рекомендована Радою
факультету прикладної математики
НТУУ "КПІ"

Надійшла до редакції
22 грудня 2010 року