

УДК 004.415.2 (045)

*О.А. АВРАМЕНКО*, канд. техн. наук, доц., НАУ;Київ

*І.С. ЯСЕНОВА*, канд. техн. наук, доц., НАУ Київ

## **ПЕРЕТВОРЕННЯ МОДЕЛЕЙ ДОКУМЕНТІВ У РЕДОКУМЕНТУВАННІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

Розглядається реалізація редокументування успадкованого програмного забезпечення, яка ґрунтується на моделі-керованому підході. Описані моделі документа на різних рівнях абстракції із врахуванням вимог технологій розробки. Запропоновано схему послідовного перетворення моделей.

Ключові слова: редокументування, моделі-керований підхід, модель документа

Рассматривается реализация редокументирования наследуемого программного обеспечения, которая основана на модели-управляемом подходе. Описаны модели документа на разных уровнях абстракции с учетом требований технологий разработки. Предложена схема последовательного преобразования моделей.

Ключевые слова: редокументирование, модели-управляемый подход, модель документа

The article describes the realization of legacy software redocumentation on the basis of model-driven approach. Software document models located at different abstract levels are proposed. They meet the requirements to software development technology. Transformation of consistent documents models is developed.

Keywords: redocumentation, model-driven approach, model of document

### **Вступ**

Реінженерія є поширеним методом переробки успадкованого програмного забезпечення (ПЗ), який дозволяє створювати нове ПЗ із використанням готових елементів та проектних рішень успадкованого [1-2]. У процесі реінженерії застосовується документація ПЗ [3], але на практиці успадковане ПЗ, найчастіше, не має документації або наявна документація є неповною і застарілою. Тому для відновлення документації застосовується редокументування, яке використовує інформацію, отриману із успадкованого ПЗ шляхом зворотної інженерії.

### **Постановка задачі**

Документація, відновлена при редокументуванні, використовується у подальшому процесі розробки ПЗ, тому повинна відповідати вимогам технології розробки, призначеній у програмному проекті [4]. Технології розробки містять, у тому числі, вимоги до складу документації та змісту документів ПЗ. Документ ПЗ можна подавати у вигляді моделі, яка формалізовано описує документ, враховуючи загальні вимоги до нього та вимоги, суттєві з точки зору редокументування. Більш того, узагальнення моделей документів технологій дозволяє побудувати їх ієрархію, на верхньому рівні якої розташовується узагальнена модель документу ПЗ. Така ієрархія надає можливість використовувати у редокументуванні поширений у програмній інженерії модельно-керований підхід (Model-Driven Development – MDD) [5], що дозволяє

зменшити витрати на створення документації при систематичному застосуванні реінженерії. У статті пропонується схема перетворення моделей документів для створення документів успадкованого ПЗ в процесі реінженерії.

### **Моделі документів в аспекті редокументування**

За результатами аналізу загальної концепції документу ПЗ та його складових, а також виходячи з властивостей документації технологій розроблення [6-8], для моделювання керування редокументування пропонуємо використовувати сукупність моделей документу ПЗ, розміщених на різних рівнях, а саме:

- модель документу ПЗ  $M_G$  на рівні загальної концепції документу ПЗ, яка описує загальні аспекти змісту і структури документу ПЗ з урахуванням редокументування незалежно від вимог технологій розробки;
- мета-модель документу технологій розробки  $M_T$  на рівні вимог технологій розроблення, яка інтерпретує модель  $M_G$  шляхом деталізації змісту і структури документу ПЗ з урахуванням загальних вимог технологій розробки  $R$ ;
- модель документу технологій розробки  $M_D$  на рівні вимог до певного документу обраної технології, яка є екземпляром мета-моделі документу технологій розробки  $M_T$ . Ця модель дозволяє описувати зміст, структуру і форму представлення конкретного документу технології як метадані (метаописи), які обумовлюють створення екземплярів документів при редокументуванні, у тому числі, за допомогою засобів автоматизації.

### **Модель документу 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 [9].

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

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

- основних ( $S_b$ ), які визначаються інформаційним змістом документу (наприклад, розділи, підрозділи та пункти документу);
- ідентифікаційних ( $S_{id}$ ), які призначені для ідентифікації документу (наприклад, назва та номер версії документу);
- загальних ( $S_g$ ), які необхідні для пошуку та навігації у документі (наприклад, глосарій та зміст документу).

Множина  $s$  утворює дерево  $G$ , яке складається з вузлів (СЕД) та ребер, які 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  $s_1 - s_n$  – СЕД першого (верхнього) рівня (зміст, розділи, додатки, предметний вказівник і т. ін.).

*Вміст* документу  $c$  представлено множиною  $C = \{c_j / j = 1..L\}$ , де  $c_j$  – інформаційний елемент документу (ІЕД) – логічно завершений інкапсульований елемент, який містить інформацію про ПЗ, наприклад, діаграма класів, опис інтерфейсів. Між елементами множин  $c$  і  $s$  визначена відповідність  $O$ : якщо задано певний ІЕД  $c \in C$ , то визначено й СЕД  $s \in S$ .  $O$  визначено для будь-якого елемента з множини  $C$  та є відображенням  $O: C \rightarrow S$ . ІЕД формується шляхом включення до нього подання, яке є інформацією про частини ПЗ або про ПЗ у цілому. При редокументуванні подання береться з успадкованого (існуючого) ПЗ, використовуючи, 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 характеризується властивостями  $p_1, \dots, p_n$  (тип, нотація, фізичне розташування значення подання):

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

Під час редокументування подання  $v_i$  з множини  $V$  можуть відповідати одному або декільком елементам  $c_j$  множини  $C$ . Відповідність множин  $v$  і  $c$  документу можна представити у вигляді матриці  $A[K, L]$ , де  $K$  – кількість елементів множини  $V$ ,  $L$  – кількість елементів множини  $C$ ,  $A = \{a_{ij} | a_{ij} = \{0, 1\}$ ,  $1$  – 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 документу утворюють множину  $EX = \{ex_i | i = 1, \dots, M\}$ .

Таким чином, для  $\forall c_k \in C$  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 документу визначається правилами зовнішнього оформлення, які відповідають стандартизованим або корпоративним стилям оформлення та залежать від формату носія документу.

### **Мета-модель документу технологій розробки програмного забезпечення**

Мета-модель документу технологій розробки ПЗ  $m_T$  створюється на основі розробленої моделі документу ПЗ  $m_G$  і вимог, спільних для технологій розробки

$R = \bigcap_{i=1}^N R_i$ , де  $R_i$  – вимоги  $i$ -ої технології. Мета-модель  $M_T = \langle C(R^c), S(R^s), P(R^p) \rangle$  під час її перетворення параметризується за трьома складовими документу: вмістом, структурою, формою представлення, на основі відповідних вимог  $R^c$ ,  $R^s$ ,  $R^p$ . Для вмісту параметром виступає нотація представлень  $p_{notation}$ . Для структури параметром виступає 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 параметром виступає множина правил форматування  $F$ , які встановлюють єдині правила оформлення для документів технологій. Таким чином, параметризована мета-модель має вид  $M_T(p_{notation}, S'_1, F)$ .

Спираючись на спільні вимоги технологій розробки, до 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 документу  $p_{name}$ .

### **Модель документу технології розробки програмного забезпечення**

Модель документу технології  $M_D$  побудована на основі моделі  $M_T$  і вимог  $i$ -ої  $R_i$  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  $p_{type}$  і пояснень  $EX$ . Для структури параметрами виступають 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  $C$  та  $V$  і відображення  $O: C \rightarrow S$ .

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

### **Редокументування як перетворення моделей**

Редокументування відповідно до моделі-керуваного підходу розглядається як процес створення множини документів  $D_{RD}$  успадкованого ПЗ через перетворення моделей  $M$  документів з формуванням їх вмісту на основі множини подань  $V$  успадкованого ПЗ. Перетворення моделей керується двома параметрами – обраною технологією розробки  $T$  і фазою  $Ph$ . Подання розташовуються у документах на основі матриці  $A[K, L]$ , яка відображає відповідність множин подань  $V$  і вмісту  $C$ , і відповідності  $O$  між елементами множин вмісту  $C$  і структури  $S$ .

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

-моделі документу  $M_G$  в мета-модель  $M_T$  документу технологій розробки ПЗ шляхом конкретизації з урахуванням вимог, загальних для всіх технологій розроблення  $R$ ;

-мета-моделі  $M_T$  у модель документів  $M_D$  технології шляхом конкретизації з урахуванням вимог обраної технології розробки  $R_i$ ;

-моделі  $M_D$  у метаописи  $m_D$  документів обраної технології розробки шляхом підстановки у параметри  $M_D$  фактичних значень, які визначаються вимогами  $R_{ij}$  до  $j$ -го документу.

Останній крок реалізації редокументування – створення документів успадкованого ПЗ  $D_{RD}$  як

екземплярів метаописів  $m_D$ .

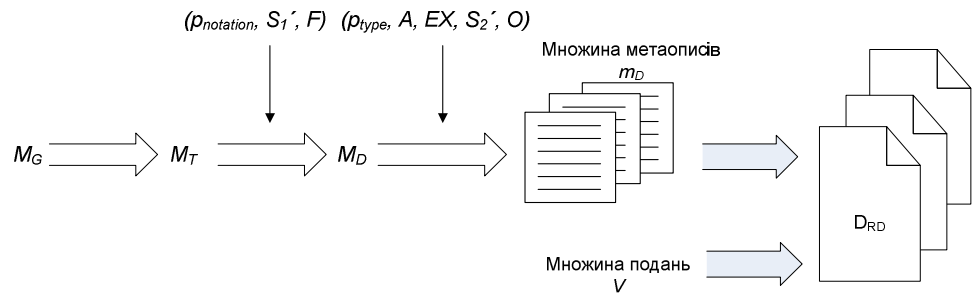


Рис.1. Схема перетворення моделей під час відновлення документів успадкованого ПЗ

### Перетворення моделей документів

Перетворення моделі документу  $M_G$  в мета-модель  $M_T$  документу технологій розробки ПЗ здійснюється шляхом відображення елементів моделі  $M_G$  в елементи моделі  $M_T$  на мові UML та конкретизації моделі  $M_G$  з урахуванням загальних вимог технологій розробки  $R$ . Якщо загальні вимоги технологій до документів не змінюються, то мета-модель  $M_T$  може бути побудована один раз та застосовуватися кожного разу при визначенні моделі  $M_D$  конкретної технології.

Відображення дозволяє отримати  $M_G$  у вигляді діаграми класів на мові UML (рис.1), використовуючи правила відображення елементів (табл.1). Структура документу в моделі  $M_G$ , представлена у вигляді дерева, відображається в ієрархію класів із суперкласом «СЕД» (рис.2).

Під час конкретизації моделі  $M_G$  на мові UML до неї додаються елементи, які описують параметри моделі  $M_T$  (див. рис.1). Параметр  $P_{notation}$  описується

перелічувальним

типом «нотація подань» для переліку можливих значень параметру.

Для опису параметра  $S_1'$  до моделі впроваджуються класи «Назва проекту», «Версія», «Дата випуску», «Автор», щоб задати множину  $S_{id}$  ідентифікаційних СЕД технологій розробки, і обмеження для опису множин ребер  $E_1' : \{x\}$ , де  $x$  – порядковий номер ребра для кореневого вузлу «Документ», та  $\{x.x\}$ , де  $x.x$  – порядковий номер ребра для абстрактного вузла «Ідентифікаційний СЕД» і вузла

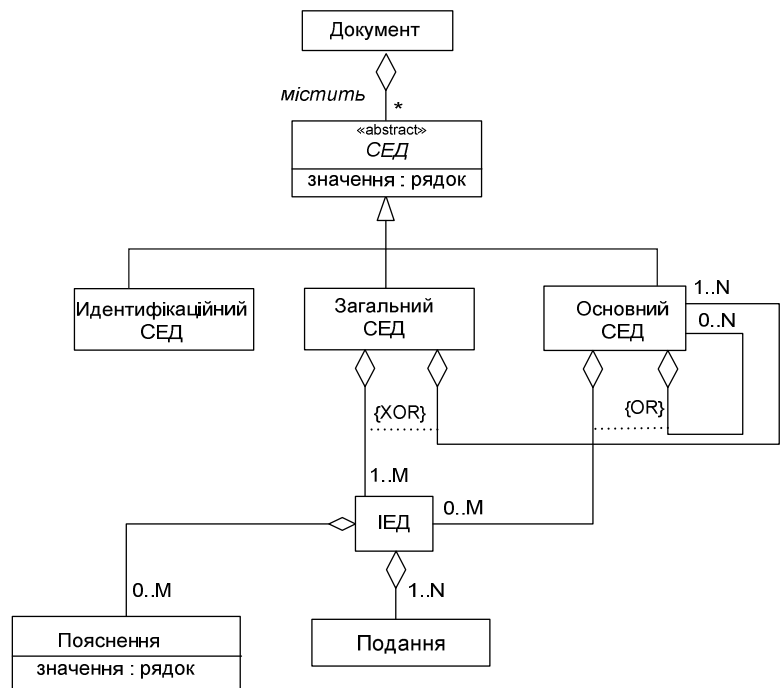


Рис.2. Модель документу ПЗ  $M_G$  на мові UML

«Загальний СЕД». Параметр  $F$  описується пакетом «Стилі форматування ( $F$ )», який включає класи для елементів оформлення документів.

Таблиця 1. Правила і приклади відображення елементів моделі  $M_G$  в елементи метамоделі  $M_T$

Правило відображення		Приклад відображення	
Елемент моделі $M_G$	Елемент мета-моделі $M_T$	Елемент моделі $M_G$	Елемент мета-моделі $M_T$
Елемент документу (ІЕД, СЕД, представлення, пояснення)	Клас	$S_b$	
Значення елемента документа (ІЕД, СЕД, пояснення)	Атрибут класу типу «рядок»	$z(ex)$	значення : рядок
Відповідність елементів моделі	Відношення агрегації («містить»)	$\rightarrow$	
Кількість елементів	Кратність асоціації	$\bigcup_{i=1}^N$	
Об'єднання неперетинних підмножин	Відношення узагальнення («це-є»)	$\cup$	
Операції «АБО», «АБО, що виключає»	Обмеження (для відношення агрегації)	$\vee$	

Отже, мета-модель документу технологій розробки ПЗ  $M_T$ , отримана внаслідок відображення та конкретизації моделі документу  $M_G$ , представлена на рис.3. Перетворення мета-моделі  $M_T$  в модель документу  $M_D$  технології здійснюється шляхом визначення значень  $S_g$ ,  $F$ , діапазону значень параметру  $p_{notation}$  і діапазонів значень властивостей документу  $P_{tech}$ ,  $P_{phase}$ ,  $P_{name}$ .

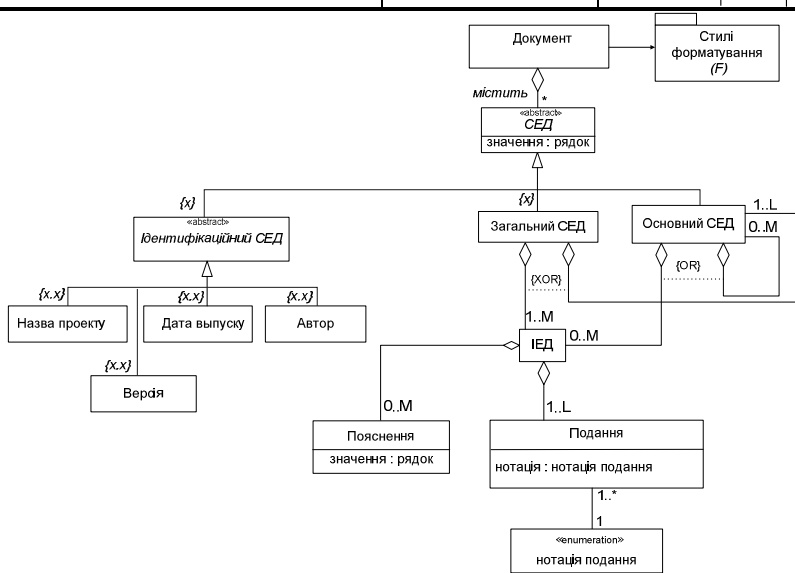


Рис.3. Мета-модель документу технологій розробки ПЗ  $M_T$

Отримана модель описує узагальнений документ конкретної технології розробки. Розглянемо перетворення моделі документів  $M_D$  технології розробки RUP. Значення параметру  $s_g$  визначається множиною службових СЕД всіх

документів технології. У технології RUP до елементів цієї множини відносяться «Зміст» і «Вступ». При побудові моделі  $M_D$  до неї додаються класи «Зміст» і «Вступ», які є підкласами абстрактного класу «Загальний СЕД». Діапазон значень параметру  $P_{notation}$  встановлюється значенням перелічуваного типу «Нотація представлення» (пакет «Властивості представлень ( $P_{notation}$ )»), який визначається вимогами технології RUP: UML, текст, таблиця.

Діапазони значень властивостей документу  $P_{tech}$ ,  $P_{phase}$ ,  $P_{name}$  встановлюються значеннями перелічуваних типів «Назва технології», «Назва фази» і «Назва документу» (пакет «Властивості документу ( $prop$ )»), які визначаються вимогами технології RUP (рис.4).

З огляду на те, що перетворення метамоделі  $M_T$  в модель  $M_D$  документа

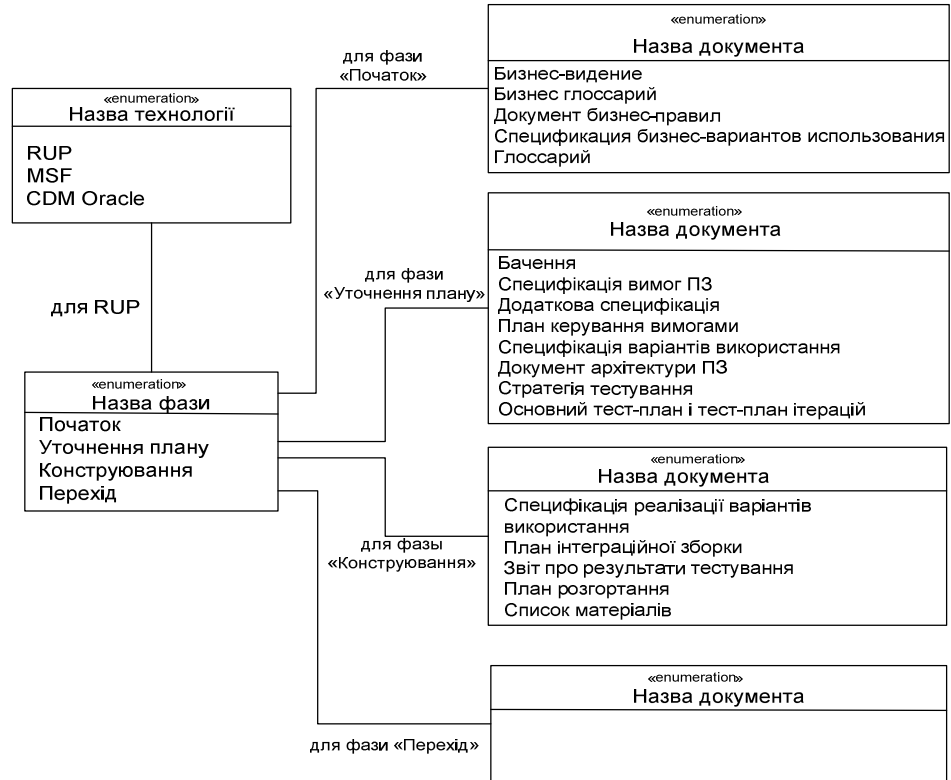


Рис.4. Значення перелічувальних типів «Назва технології», «Назва фази» і «Назва документу» для технології RUP

технології здійснюється з урахуванням вимог певної технології розробки, то за умови незмінності вимог технології до документів, модель  $M_D$  може бути побудована лише один раз і використовуватися кожного разу під час створення метаопису  $m_D$  конкретного документа технології.

Перетворення моделі  $M_D$  в метаопис  $m_D$  документів обраної технології розробки  $R_{ij}$  здійснюється шляхом підставлення фактичних значень параметрів у модель  $M_D$ , які визначаються на основі вимог  $R_{ij}$  технології до певного свого документу. Метаопис  $m_D$  документа являє собою екземпляр моделі  $M_D$ . Засобом побудови метаописів є спеціально розроблена шляхом розширення мови XML мова Redocumentation Document Description Language (RDDDL) [10]. Мова RDDDL містить засоби для опису СЕД, ІЕД, представлень, пояснень та властивостей документу.

Для документів однієї технології розробки перетворення моделей виконуються одноразово і документи створюються на основі готових метаописів, використовуючи для цього процеси трансляції та збирання, які можуть бути автоматизовані. Для автоматизації пропонується архітектура засобів [11], які відносяться до CARSE та виконують наступні функції: підготовку метаописів

документів для редокументування  $m_D$ , підготовку представлень  $V$  успадкованого ПЗ та створення документів  $D_{RD}$ .

### **Висновки**

Застосування модельно-керованого підходу при редокументуванні дозволяє створювати документи ПЗ шляхом послідовного перетворення моделей від абстрактного рівня до реалізації, що надає можливість забезпечувати вимоги технологій розробки до документації та спрощує редокументування за рахунок систематичного повторного використання моделей. Запропоновані схема та правила перетворення моделей документів дозволяють формалізувати процес створення документів та параметризувати його вимогами обраної технології розробки.

**Список літератури:** 1. *Chikofsky E. J., Cross II J. H. Reverse Engineering and Design Recovery: A Taxonomy // IEEE Software. – 1990. – January. – P.13-17* 2. *Сидоров Н.А. Восстановление, повторное использование и переработка программного обеспечения. I. // УСиМ. – 1998. – № 3. – С. 74-83.* 3. *Сидоров Н.А., Хоменко В.А., Авраменко Е.А. Реинженерия проектов программного обеспечения. // Проблемы програмування. – 2006. – № 2,3. — С. 31–38.* 4. *Сидоров Н.А., Авраменко Е.А. Метод и средства редокументирования наследуемого программного обеспечения // Проблемы програмування. – 2008. – № 2,3. – С. 229-238* 5. *Mellor S.J., Clark A.N., Futagami T. Model-Driven Development // IEEE Software. – 2003. – Vol.20, No.5. – P.14-18.* 6. *Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения : Пер. с англ. – СПб.: Питер, 2002. – 496 с.* 7. *Microsoft Solutions Framework. Модель процессов MSF. Версия. 3.1. – Корпорация Майкрософт (Microsoft Corp.), 2002. – 44 с.* 8. *Oracle Method®. CDM Classic Method Handbook. Release 2.6.0. – Oracle corp., 2000. – 248 p.* 9. *ДСТУ 4302:2004 (ISO/IEC 6592:2000(MOD)) Інформаційні технології. Настанови щодо документування комп'ютерних програм. – К: Держпозживстандарт, 2005. – 30с.* 10. *Авраменко Е.А. Средство редокументирования наследуемого программного обеспечения // Вісник Національного технічного університету України „КПІ”. Серія „Інформатика, управління та обчислювальна техніка” : зб. наук. праць. – Київ : „ВЕК+”, 2007. – № 47. – С. 100–107.* 11. *Авраменко О.А. Архітектура засобів редокументування успадкованого програмного забезпечення. – Вісник НАУ. – 2007. – № 3-4 (33). – С. 58-62*

*Поступила в редколлегию 19.03.2011*

### **УДК 332.339**

**М. О. БІЛАН**, асп., Черкаський національний університет ім.  
Б.Хмельницького

### **РОЗРОБКА МОДЕЛІ ВПРОВАДЖЕННЯ БІЗНЕС-ПРОЦЕСІВ У ГОТЕЛЬНОМУ ГОСПОДАРСТВІ**

Запропонована модель впровадження та використання сервісно-орієнтованої системи управління бізнес-процесами підприємств готельного господарства, яка дозволяє реалізувати автоматизовані дії та надає можливість систематично та ефективно використовувати існуючі ресурси.

Ключові слова: система управління бізнес-процесами, сервісно-орієнтована архітектура, нотація BPMN.

Предложена модель внедрения и использования сервисно-ориентированной системы управления бизнес-процессами предприятий гостиничного хозяйства, которая позволяет