УДК 004.314:338.5

МЕТОД И СРЕДСТВА РЕДОКУМЕНТИРОВАНИЯ НАСЛЕДУЕМОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Н.А. Сидоров, Е.А. Авраменко

Национальный авиационный университет,
03058, проспект Комарова, 1.
Тел.: 406 7396, факс (044) 497 8106,
e-mail: sna@nau.edu.ua

Рассматривается метод редокументирования наследуемого программного обеспечения, ориентированный на технологию разработки. Предлагаются языки описания документов и представлений наследуемого программного обеспечения, архитектура средств для реализации метода редокументирования. Приводятся результаты практического применения метода и средств при проведении редокументирования одного из проектов программного обеспечения.

This article considers the technology oriented method of redocumentation of legacy software. There are proposed the languages of description of documents and views, the architecture of facilities for realization of the redocumentation method. The results of practical application of methods and facilities in one of software project are provided.

Введение

Задача редокументирования программного обеспечения (ПО) возникла в 80-х годах прошлого столетия в связи с возросшим количеством наследуемого ПО и проектов, связанных с его реинженерией [1–4]. К концу 80-х годов в таких проектах были задействованы 40 % программистов, в 2000 году – 60 % [5]. Эффективность реинженерии зависит от полноты и качества документации наследуемого ПО. Чаще всего такое ПО не имеет документации или она устарела. При реинженерии требуется редокументирование для создания документации наследуемого ПО.

Редокументирование ПО

Определение редокументирования ПО рассматривается в [1–4]. В работе [2] редокументирование определяется как создание или пересмотр семантически эквивалентных представлений внутри одного абстрактного уровня. В качестве результата редокументирования выступают альтернативные формы представления, ориентированные на определенных пользователей [1, 2].

В работе [3] рассматривается структурное редокументирование, сущность которого заключается в восстановлении архитектурных аспектов ПО с использованием обратной инженерии [3, 6]. Результатом являются многоуровневые иерархические представления, составляющие основу документации ПО, которые формируются экспертом на основе извлеченной из исходного кода информации.

В работе [4] редокументирование определяется как процесс анализа системы, результатом которого является документация, представленная в различных формах, включая руководства пользователей и реформатированные листинги исходных кодов.

В работе [7] описывается инкрементный подход к редокументированию на основе Веб-технологий. Заданная и изменяемая структура аннотаций в форме гипертекста заполняется информацией, извлеченной из исходного кода, и дополняется комментариями самого разработчика ПО.

Средство ARIS Redocumentation Scout (RDS) SAP предназначено для редокументирования бизнес-процессов, поддерживаемых ПО компании SAP, и обеспечивает поиск, сбор и обработку информации о незадокументированных бизнес-процессах, реализуемых в SAP-системе [8].

Основное отличие редокументирования от документирования состоит в том, что первое реализуется в рамках обратной инженерии ПО. Объектом редокументирования является наследуемое ПО, а объектом документирования – разрабатываемое ПО [9].

Метод редокументирования, ориентированный на технологию разработки

Предлагаемый метод предназначен для использования в реинженерии наследуемого ПО. Реинженерия ПО включает два типа процессов – обратной и прямой инженерии. Процессы обратной инженерии обеспечивают восстановление информации из наследуемого ПО. Процессы прямой инженерии – это процессы технологии разработки ПО. Поэтому документация, создаваемая при редокументировании и используемая в прямой инженерии, должна соответствовать требованиям технологии, выбранной для прямой инженерии. Это составляет сущность предлагаемого метода редокументирования (рис. 1).

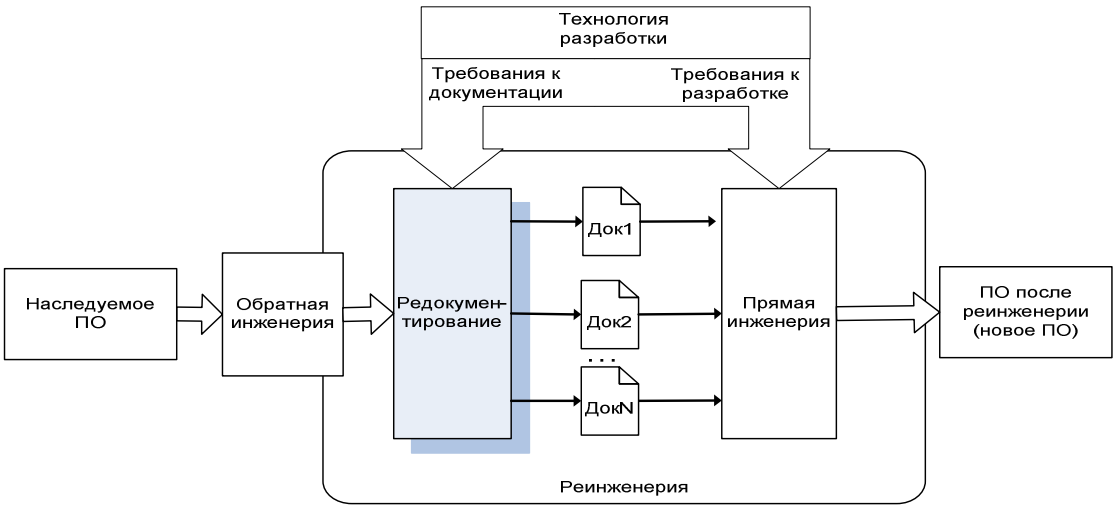

Рис.1. Метод редокументирования наследуемого ПО, ориентированный на технологию разработки

«Глубина» проведения реинженерии наследуемого ПО зависит от фазы, с которой начинается прямая инженерия. Поскольку каждая фаза прямой инженерии опирается на документы предыдущей фазы, то при редокументировании необходимо создать документы фазы, предшествующей той, с которой начинается прямая инженерия (рис. 2). Таким образом, процесс редокументирования в предлагаемом методе определяется выбранной технологией разработки ПО и фазой, начиная с которой предполагается проводить его разработку.

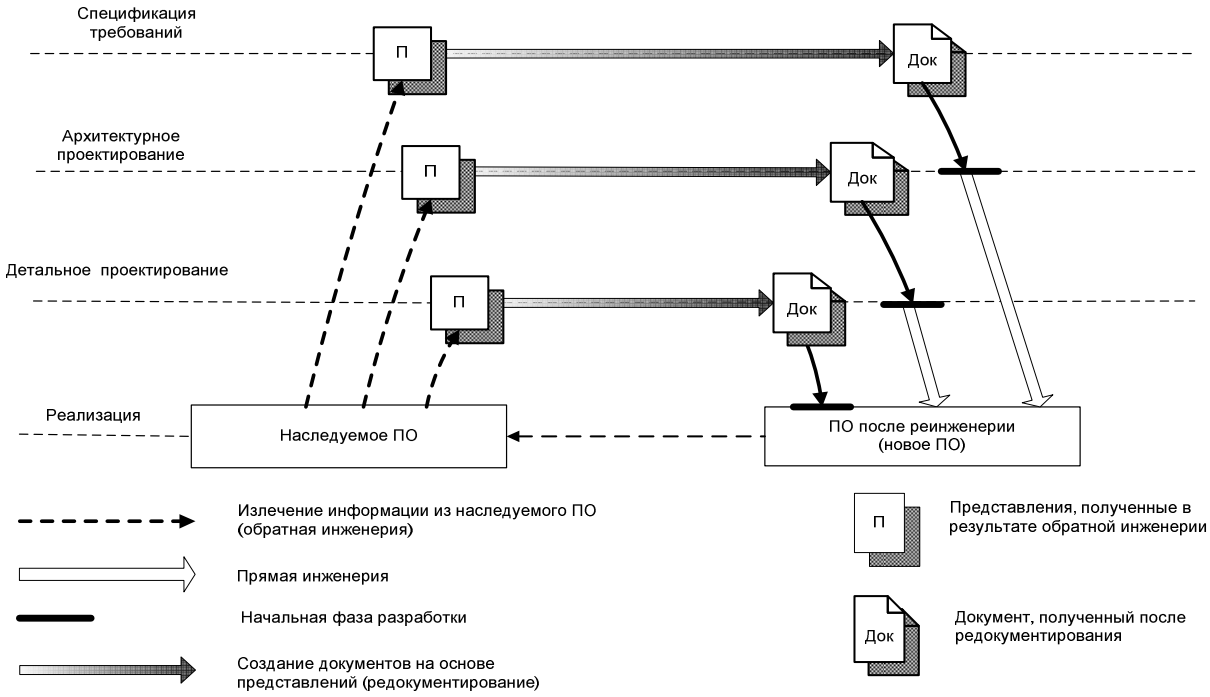

Рис. 2. Редокументирование наследуемого ПО в зависимости от начальной фазы разработки

Модель документа

Документ – это материальный объект, содержащий информацию в фиксированном виде, специально предназначенный для передачи во времени и пространстве [10]. В отраслевых стандартах документ ПО определяется как уникально идентифицируемая единица информации, разработанная для определенного круга пользователей, с определенной целью, и записанная на любом носителе информации [11–13].

Документ ПО D можно представить тройкой $D =< S, C, P >$, где S – структура документа, C – содержимое документа (информационное наполнение), P – внешнее представление документа.

Структура документа S – это множество $S = \{s_i | i = 1..M\}$, где s_i – структурные элементы документации (СЭД), которые можно рассматривать как объединение $S = S_b \cup S_{id} \cup S_g$ трех непересекающихся

подмножеств СЭД:

– основных (S_b), определяемых информационным содержимым документа (например, главы, параграфы и пункты документа);

– идентификационных (S_{id}), предназначенных для идентификации документа (например, название и номер версии документа);

– общих (S_g), необходимых для поиска и навигации в документе (например, глоссарий и содержание документа).

Множество S образует дерево G, состоящее из узлов (СЭД) и ребер, отображающих отношение иерархической упорядоченности между СЭД:

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

где E – множество линейно упорядоченных ребер дерева.

Основные и общие СЭД могут содержать вложенные СЭД, порядок расположения которых задается списком ребер. Например, для корневого СЭД дерева s_d список ребер имеет вид:

$$\left((s_d, s_1), (s_d, s_2), ..., (s_d, s_i), ..., (s_d, s_j), (s_d, s_n)\right).$$

Содержимое документа C представлено множеством $C = \{c_j \mid j = 1..N\}$, где c_j – информационный элемент документа (ИЭД) – логически завершенный инкапсулированный элемент, содержащий информацию о ПО, например, диаграммы классов, объектов, описание интерфейсов.

Между элементами множеств C и S определено такое соответствие R, что если задан некоторый ИЭД $c \in C$, то определен и СЭД $s \in S$. R определено для любого элемента из множества C и является отображением $R: C \to S$.

При редокументировании ИЭД формируется путем включения в него представления, которое извлекается из наследуемого ПО, и является информацией о части ПО или ПО в целом. Примерами представлений ПО являются перечень требований, диаграммы вариантов использования, компонентов и классов, описание класса, изображение пользовательского интерфейса, схема базы данных (БД). Представление v имеет значение z (текст, рисунок, таблица, диаграмма, анимация или звук) и характеризуется свойствами $p_1,..., p_n$ (тип, нотация, физическое размещение значения представления):

$$v = \langle z, p_1, ..., p_n \rangle.$$

ИЭД, кроме представления, может включать пояснение, например, примечание, подписи таблиц и рисунков.

Форма представления документа определяется правилами внешнего оформления, которые отвечают стандартизированным или корпоративным стилям оформления [11, 14] и зависят от формата носителя документа.

Документация технологий разработки

Рассмотрим современные технологии разработки RUP, MSF, CDM Oracle и гибкие технологии [15–18]. Документы технологий разработки состоят из документов фаз. Будем учитывать требования технологий к составу документации, структуре и отдельных содержанию документов.

Документы технологий относятся к одной из трех категорий: разработки, продукта и процесса [19]. Состав документации регламентируется перечнем документов для каждой категории (рис. 3). Документ в перечне характеризуется названием, кратким описанием и указанием фазы, к которой он относится.

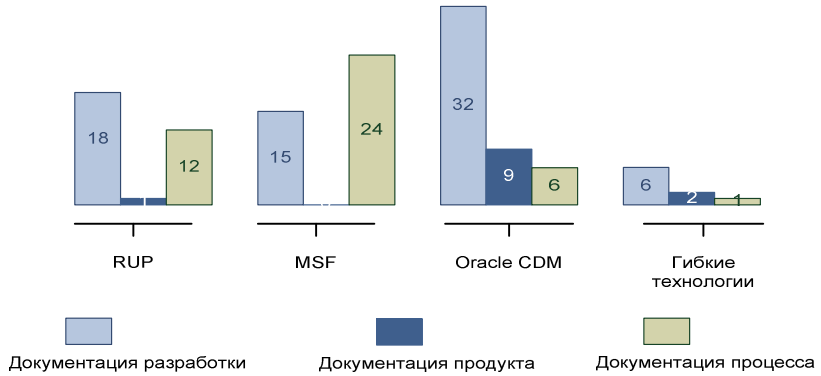

Рис. 3. Распределение документов технологий по категориям

Технологии RUP [15] и MSF [16] используют перечень документации разработки и процесса с детальным описанием содержания каждого из документов, а документация продукта рассматривается как часть поставки ПО пользователю. Технология CDM [17] использует полный перечень документации разработки и продукта с детальным описанием содержания каждого из документов, а документацию процесса не регламентирует. В гибких технологиях состав документации – это список рекомендуемых документов, который может применяться полностью или частично в зависимости от требований заказчика и условий проекта (объем, сложность, стоимость, ожидаемое время жизни проекта, характер взаимодействия с другими командами разработчиков). В гибких технологиях управление процессом разработки осуществляется путем личного общения участников, поэтому документация процесса не используется. В работе создание документов продукта и процесса не рассматривается.

Структура и содержание документов технологий представляются следующими формами: общая характеристика документа, перечень пунктов содержания с их описанием, шаблон и образец документа. В технологиях разработки применяются одна из перечисленных форм или их комбинации (табл. 1). Общая характеристика документа – это сжатое неструктурированное описание содержания документа, которое указывает на те аспекты, которые должен содержать документ. Перечень пунктов содержания документа – это формализованный список пунктов содержания, которые должны быть раскрыты в документе. Каждый пункт может иметь характеристику, позволяющую уточнить его содержимое. Шаблон – это формализованное описание структуры и содержания документа в виде предопределенных структурных элементов и описаний их содержимого. Образец документа – это результат правильного заполнения шаблона документа в одном из проектов. Он позволяет создавать документы по аналогии.

Таблица 1

Технология разработки	Форма представления структуры и содержания документа			
	Общая характеристика документа	Перечень пунктов содержания	Шаблон документа	Образец документа
RUP	+	–	+	+
MSF	–	–	+	–
CDM	+	+	–	–
Гибкие технологии	+	–	–	–

Реализация метода

Состав документации при редокументировании определяется фазой, начиная с которой проводится прямая инженерия ПО. Структура и содержание документов определяются требованиями выбранной технологии разработки. Содержимое документов формируется на основе результатов обратной инженерии.

Реализация метода состоит в следующем. Для каждой технологии создаются комплекты метаописаний документов, относящихся к фазам (рис. 4). Представления, извлеченные из наследуемого ПО с помощью обратной инженерии, снабжаются дескриптором и размещаются в хранилище представлений. По метаописанию и представлению для заданных технологии и фазы генерируется документ.

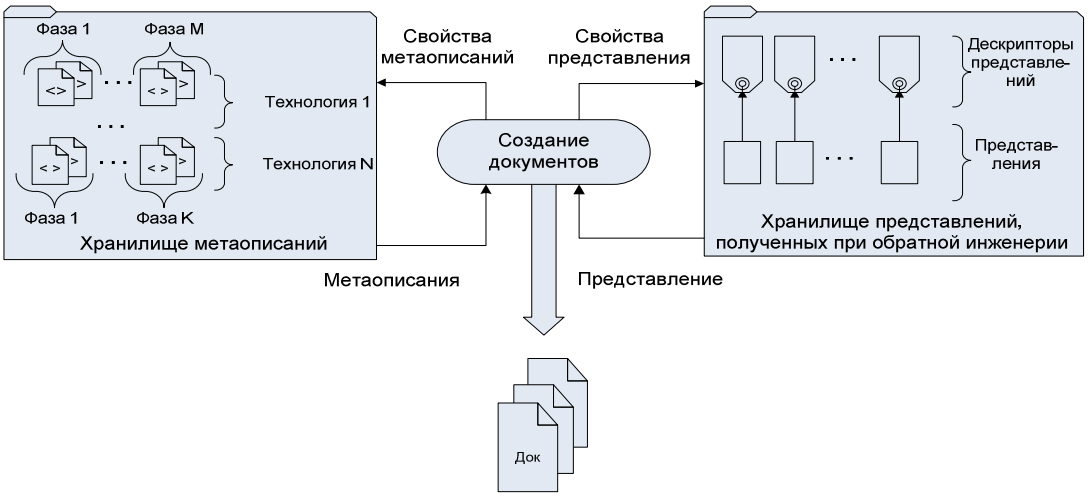

Рис. 4. Схема использования метаописаний при редокументировании

Реализация метода предусматривает применение метаописаний документов (рис. 4), которые строятся экспертом в соответствии с моделью документа путем выявления ИЭД, СЭД и их расположения в документе. Кроме того, в метаописании указываются свойства представлений, необходимых для формирования содержимого документа. Применение метаописаний позволяет автоматизировать процессы создания документов и отказаться от использования текстовых редакторов (рис. 5). Созданный экспертом комплект метаописаний документов технологии может использоваться при проведении редокументирования повторно.

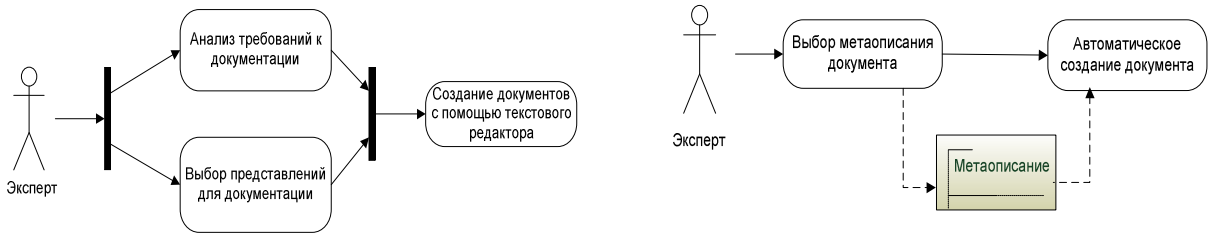

Рис. 5. Создание документов при редокументировании:
а) без использования метаописаний; б) с использованием метаописаний

Для описания документов используются языки – Redocumentation Document Description Language (RDDL) и Redocumentation View Description Language (RVDL), разработанные на основе расширенного языка разметки XML [20]. RDDL содержит средства для описания СЭД, ИЭД, представлений, пояснений и свойств документа (табл. 2, 3).

Таблица 2

Элемент документа	Назначение элемента	Метка
Документ	Корень описания документа	document
Идентификационный СЭД	Название проекта	projectName
	Название документа	title
	Номер версии документа	version
	Лист изменений	revisionHistory
Общий СЭД	Содержание	tableofContents
	Глоссарий	glossary
	Введение	introduction
	Список иллюстраций	listOfFigures
	Список таблиц	listOfTables
	Указатель	index
	Список литературы	bibliography
	Приложение	appendix
Основной СЭД	Структура документа	item
ИЭД	Содержимое документа	infoItem
Представление	Представление, извлеченное с помощью средств обратной инженерии	view
Пояснение	Дополнительная информация	explanation

Таблица 3

Элемент, содержащий атрибут	Атрибут	Назначение атрибута
Документ	technology	Технология разработки
	phase	Фаза разработки
Представление	type	Тип представления
	notation	Нотация представления
Основной СЭД	iterative	Повторение элемента

RVDL предоставляет метки для корня дескриптора (descriptor) и свойств представления: тип (type), нотация (notation), название проекта (project), имя (fileName) и тип файла (fileType). При редокументировании поиск необходимых представлений осуществляется путем сравнения значений атрибутов type и notation метаописания и значений элементов type и notation дескриптора. При включении представления в документ используются значения элементов fileName и fileType. На рис. 6 показан пример дескриптора представления на языке RVDL.

```
<?xml version="1.0" encoding="utf-8"?>
<descriptor>
        <type>ListOfPackages</type>
        <notation>UML</notation>
        <project>SkyPassword</project>
        <fileName>C:\SkyPassword\Packages</fileName>
        <fileType>jpg</fileType>
</descriptor>
```

Рис. 6. Пример дескриптора представления

На рис. 7 показан пример фрагмента метаописания документа технологии RUP «Архитектура ПО», который относится к фазе «Elaboration».

```
<?xml version="1.0" encoding="utf-8"?>
<document technology="RUP" phase ="Elaboration">
        <projectName></projectName>
        <title>Software Architecture Document</title>
        <version></version>
        <revisionHistory></revisionHistory>
        ...
        <item> 5. Logical View
                <infoitem>
                <explanation>
                A description of the logical view of the architecture. Describes the most important classes, their organization in
                service packages and subsystems, and the organization of these subsystems into layers. Also describes the most
                important use-case realizations, for example, the dynamic aspects of the architecture. Class diagrams may be
                included to illustrate the relationships between architecturally significant classes, subsystems, packages and layers.
                </explanation>
                </infoItem>
                <item> 5.1 Overview
                        <infoItem>
                        <view type="ListOfPackages" notation="UML"></view>
                        <explanation>Package and Subsystem Layering </explanation>
                        </infoItem>
                </item>
        </item>
        ...
</document>
```

Рис. 7. Пример метаописания документа

Средства редокументирования

Рассматриваются три следующие фазы редокументирования (рис. 8):

- подготовка метаописаний документов для редокументирования;
- подготовка представлений наследуемого ПО;
- создание документов выбранной технологии и фазы разработки.

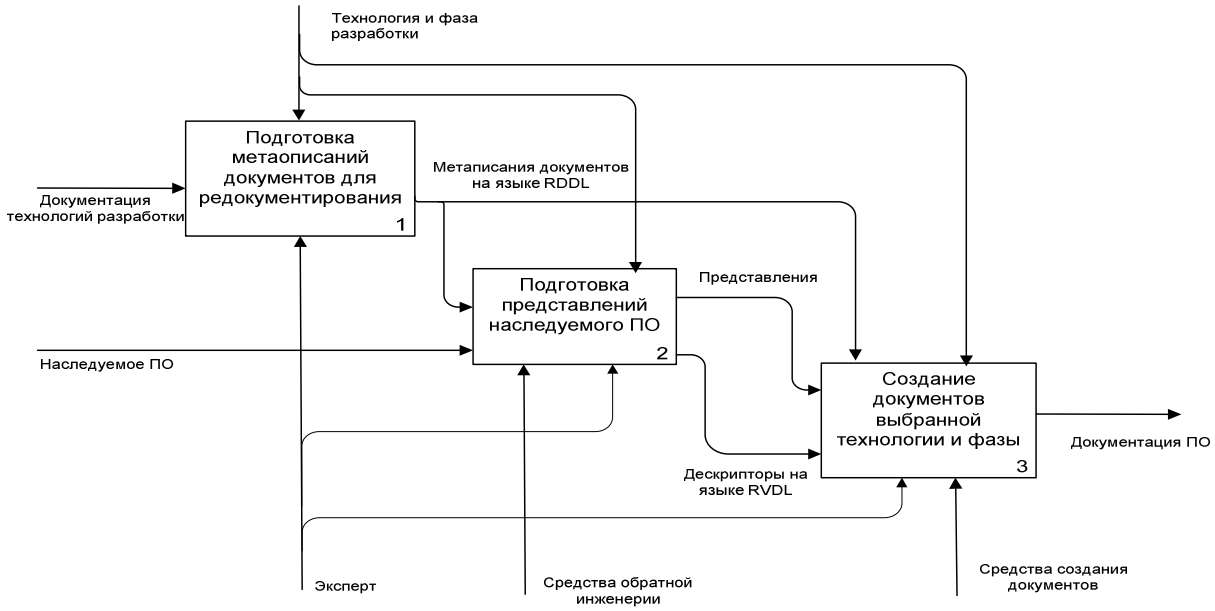

Рис. 8. Общая схема работы средств редокументирования

Для реализации указанных фаз разработаны средства редокументирования (рис. 9).

Интерфейс эксперта обеспечивает взаимодействие эксперта со средствами редокументирования в диалоговом режиме.

Подсистема управления обрабатывает запросы, поступающие от интерфейса эксперта, и обеспечивает согласованную работу средств редокументирования.

Редактор метаописаний документов выполняет такие функции:

– создание и редактирование метаописаний документов на языке RDDL – выполняется экспертом на основе документации технологий разработки;

– проверка правильности метаописаний с помощью DTD [20].

Менеджер выделения представлений используется во второй фазе редокументирования и выполняет следующие функции:

– выбор и запуск средств обратной инженерии;

– автоматическое или автоматизированное создание дескрипторов представлений;

– сохранение представлений и их дескрипторов.

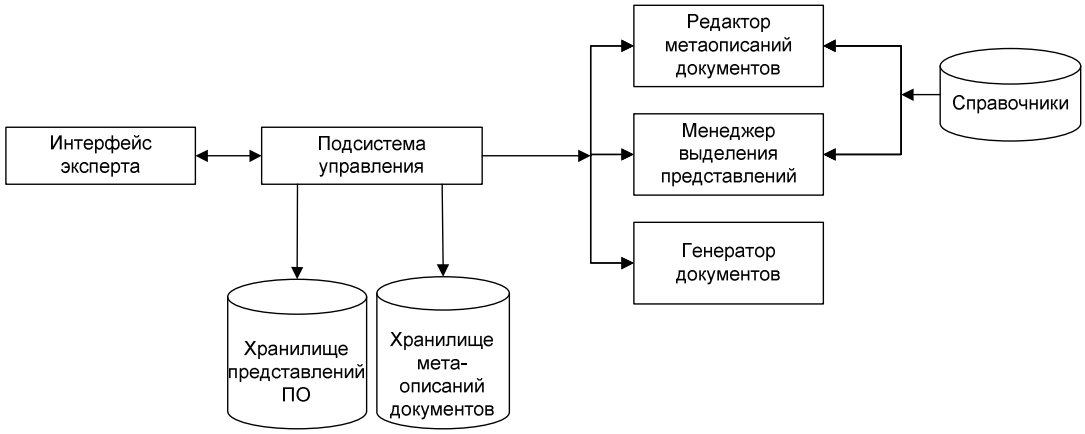

Рис. 9. Архитектура средств редокументирования

Генератор документов создает документы, заполняя их содержимое представлениями из хранилища представлений (рис. 10).

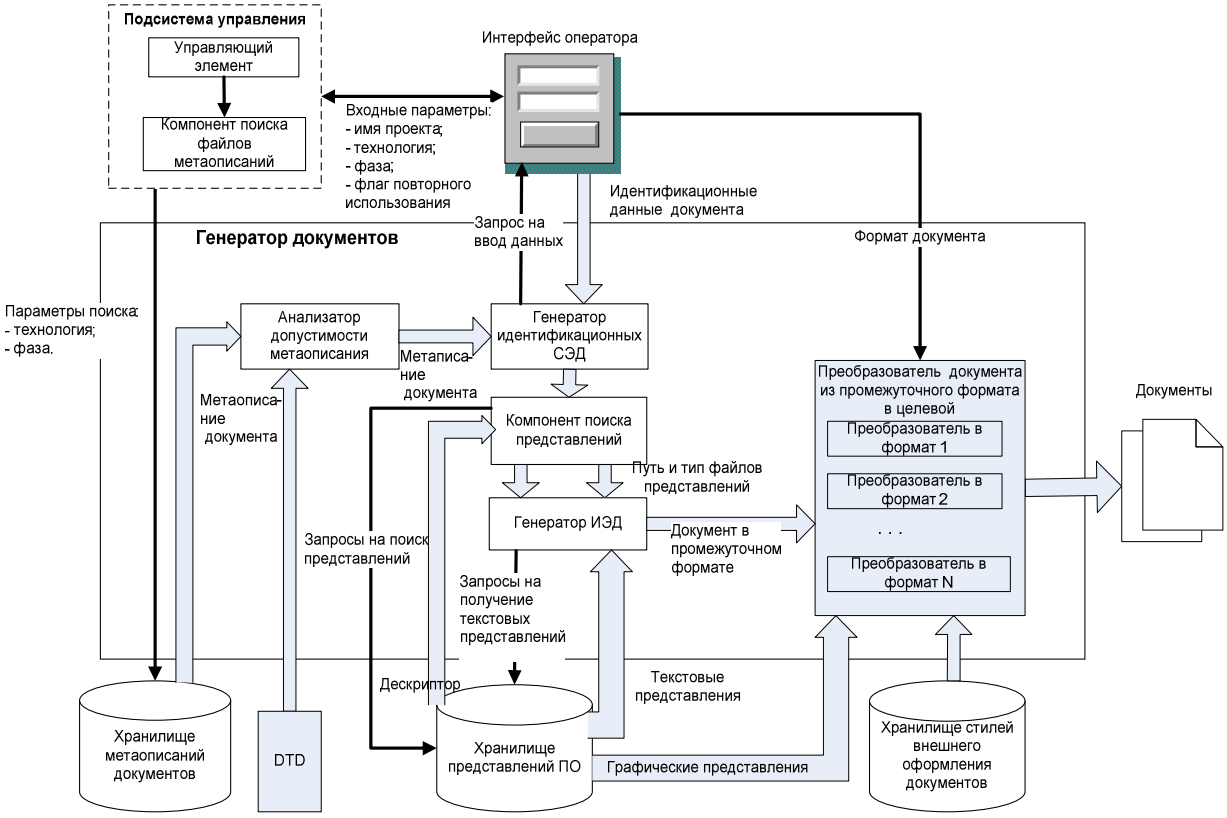

Рис. 10. Структура генератора документов

Особенностью генератора является предварительная генерация документа в промежуточном формате, который затем может быть преобразован в любой конечный формат с помощью специальных преобразователей. Генератор документов выполняет следующие функции:

– выбор экспертом технологии, фазы разработки, формата восстанавливаемых документов, для которых проводится редокументирование;

– проверка правильности каждого из метаописаний, используемых при генерации документов, с помощью DTD;

– автоматическая генерация структур документов;

– автоматическое заполнение структур документов выделенными представлениями;

– форматирование документов;

– создание комплекта документов, соответствующих выбранной технологии и фазы разработки.

Практический пример применения метода

Разработанный метод и средства редокументирования ПО были использованы при реализации задачи создания документации программно-аппаратного комплекса для изготовления документов государственного образца [21]. Создаваемая документация должна соответствовать требованиям технологии разработки RUP. Цель проведения редокументирования ПО состояла в обеспечении процесса сопровождения, поэтому восстановлению подлежали все документы, начиная от фазы «Начало» до «Переход».

Для проведения редокументирования предоставлены существующая документация, оформленная в соответствии с ГОСТ 19 серии [22], программные модули (около 15000 строк код на языке Object Pascal) и БД (60 таблиц в формате InterBase).

Для создания представлений для восстанавливаемых документов использовались два способа:

– экспертный – путем ручного анализа входных документов наследуемого ПО;

– автоматизированный – с использованием следующих средств обратной инженерии: IBM Rational Rose (средство визуального моделирования), Ensemble Rose Delphi Link 3 (программный модуль интеграции моделей IBM Rational Rose со средой разработки Delphi), IBDataWorks (средство фирмы SoftMosis для моделирования БД под SQL-серверами InterBase и Firebird).

Для оценки эффективности редокументирования использована система метрик, построенная с использованием метода GQM [23]. Цель исследования сформулирована следующим образом: «Оценить эффективность использования в практическом примере метода и средств редокументирования, ориентированных на технологию разработки». Вопросы, на которые должны быть получены ответы в результате исследования сформулированы следующим образом: «Насколько полно восстановлена документация?» и «Какова достигнутая степень автоматизации редокументирования?». Метрики, определенные для исследования результатов редокументирования, представлены в табл. 4.

Таблица 4

Цель	Вопрос	Метрика	
		Семантика	Обозначение
Оценить эффективность применения метода и средств редокументирования	Насколько полно восстановлена документация?	Количество восстановленных документов	КВД
		Количество СЭД в метаописании документа	КСЭДМ
		Количество требуемых СЭД в документах проекта	КТСЭД
		Количество восстановленных СЭД	КВСЭД
		Количество представлений в метаописании	КПМ
		Количество требуемых представлений в проекте (с учетом КТСЭД)	КТП
		Количество восстановленных представлений	КВП
		Процент восстановленных документов	ПВД
		Процент восстановленных СЭД	ПВСЭД
		Процент восстановленных представлений	ПВП
	Какова степень автоматизации редокументирования?	Процент автоматически восстановленных документов	ПАВД
		Процент автоматически восстановленных СЭД	ПАВСЭД
		Количество автоматически восстановленных представлений	КАВП
		Процент автоматически восстановленных представлений	ПАВП

В табл. 5 приведены значения метрик для каждого документа и фаз RUP, полученных по результатам редокументирования.

Таблица 5

Фаза RUP	Документ	Значения метрик						
		КСЭДМ	КТСЭД	КВСЭД	КПМ	КТП	КВП	КАВП
Начало	Бизнес-видение	21	30	30	19	28	24	0
	Бизнес глоссарий	5	30	30	4	26	26	0
	Документ бизнес-правил	4	7	7	3	6	6	0
	Спецификация бизнес-вариантов использования	22	53	53	13	44	40	0
	Глоссарий	5	15	15	4	13	13	0
Итого по фазе:	**документов: 5**	**57**	**135**	**135**	**43**	**117**	**109**	**0**

Окончание табл. 5

Фаза RUP	Документ	Значения метрик						
		КСЭДМ	КТСЭД	КВСЭД	КПМ	КТП	КВП	КАВП
Уточнение плана	Видение	37	55	55	33	51	43	1
	Спецификация требований ПО	6	6	6	30	30	27	0
	Дополнительная спецификация	22	36	36	29	43	41	1
	План управления требованиями	17	17	17	13	13	0	0
	Спецификация вариантов использования	18	23	23	23	28	19	0
	Документ архитектуры ПО	15	15	15	25	25	22	10
	Стратегия тестирования	22	22	22	30	30	1	0
	Основной тест-план и тест-план итераций	47	47	47	53	53	2	0
Итого по фазе:	**документов: 8**	**184**	**221**	**221**	**236**	**273**	**155**	**22**
Конструирование	Спецификация реализации вариантов использования	2	2	2	2	2	2	0
	План интеграционной сборки	2	2	2	7	7	3	0
	Отчет о результатах тестирования	5	5	5	6	6	0	0
	План развертывания	11	11	11	10	10	4	0
	Список материалов	8	8	8	8	8	4	1
Итого по фазе:	**документов: 5**	**28**	**28**	**28**	**33**	**33**	**13**	**1**
Переход	–	–	–	–	–	–	–	–
Всего по проекту	**документов: 18**	**269**	**384**	**384**	**312**	**423**	**277**	**23**

На рис. 11 показаны диаграммы, иллюстрирующие полученные значения метрик для представлений, распределенные по фазам RUP.

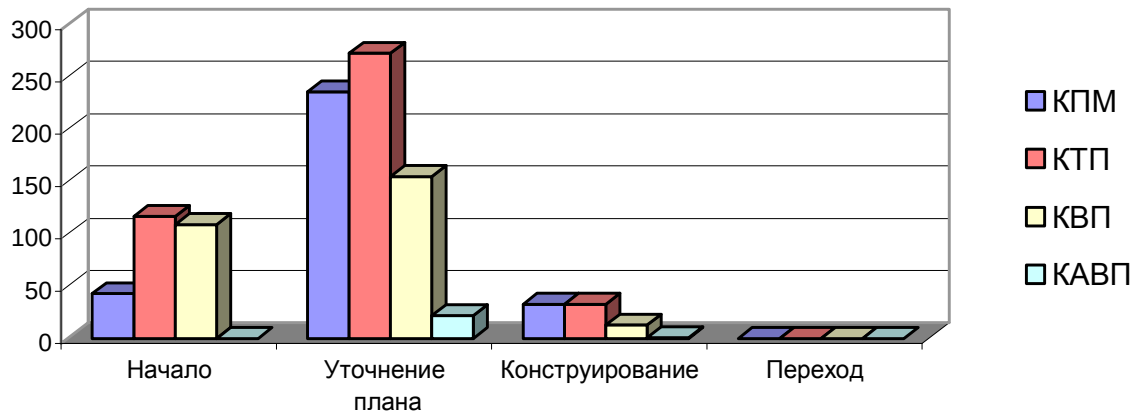

Рис. 11. Значения метрик представлений

Полученные значения метрик показывают, что количество требуемых представлений (КТП) в проекте превышает количество представлений в метаописаниях документов (КПМ). Это объясняется тем, что определенные представления, требуемые метаописанием, при редокументировании замещаются множеством однотипных представлений, например представление «спецификация класса» заменяется в документе спецификациями всех классов проекта.

В табл. 6 приведены значения метрик, позволяющие оценить эффективность восстановления документации и степень автоматизации в процентных отношениях. Анализ полученных значений метрик показывает, что редокументирование позволило автоматически восстановить 100 % документов из необходимого множества и 100 % СЭД. Это обусловлено формальным подходом метода к генерации документов и их структур – создаются все документы, для которых существуют метаописания.

Процент автоматически восстановленных представлений (ПАВП) составляет менее 10 %, что обусловлено ограниченными возможностями использованных средств обратной инженерии в условиях данного проекта: наследуемое ПО разработано на объектно-ориентированном языке с использованием структурного подхода. Наибольший процент автоматически восстановленных представлений приходится на документы фаз «Уточнение плана» и «Конструирование», которые содержат представления, отражающие исходный код программ, например, диаграмма классов или диаграмма компонентов. Такие представления восстанавливаются автоматически средствами обратной инженерии. Высокий процент восстановления представлений (ПВП) был достигнут за счет наличия документации наследуемого ПО, к которой применялся экспертный анализ.

Таблица 6

Фаза RUP	Значения метрик					
	ПВД, %	ПВСЭД, %	ПВП, %	ПАВД, %	ПАВСЭД, %	ПАВП, %
Начало	100	100	93,16	100	100	0
Уточнение плана	100	100	56,78	100	100	14,19
Конструирование	100	100	39,39	100	100	7,69
Переход	–	–	–	–	–	–
Всего по проекту	100	100	64,48	100	100	8,30

Была проведена оценка временных затрат по фазам редокументирования (рис. 8). Подготовка метаописаний документов с помощью разработанного редактора метаописаний заняла 20 чел. часов. Общее время на подготовку представлений составило около 12 чел. часов. Это время включает автоматическое формирование списка необходимых представлений – 1 мин., автоматическое извлечение представлений выбранными средствами – около 30 минут и создание дескрипторов – 1 чел. час. Время на создание документов с помощью генератора документов (рис. 10) составило 10 мин. Общее время на редокументирование (около 32 чел. часов) сопоставимо с оценкой временных затрат на проведение такого же редокументирования без применения предлагаемого метода и средств, но поскольку большую долю времени составляла подготовка метаописаний, которая выполняется только один раз для каждой технологии, то все последующие затраты на редокументирование должны сократиться более чем вдвое.

1. *Sneed H. M.* Software renewal: a case study // IEEE Software. – 1984.– 1(3). – P. 56–63.
2. *Chikofsky E. J., Cross J. H.* Reverse engineering and design recovery: a taxonomy // IEEE Software. – 1990. – 7(1). – P.13–17
3. *Tilley S.R., Smith D.B.* Perspectives on Legacy System. – Reengineering.Pittsburgh, PA: Reengineering Center, Software Engineering Institute, Carnegie Mellon University, 1995. – 146 p.
4. *Technical report.* Software Reengineering Assessment Handbook (SRAH). Version 3.0. JLC-HDBK-SRAH, March 1997. – 234 p.
5. *Jones C.* Applied Software Measurement. – McGraw-Hill, second edition, 1996. – 325 p.
6. *Wong K., Tilley S.R., Müller H. A., Storey M. D.* Structural redocumentation: a case study // IEEE Software. – 1995.– 11(6). – P. 501–520
7. *Rajlish V.* Incremental redocumentation using the web // IEEE Software.– 2000. –17(5).– P. 2–6.
8. *ARIS SAP* - http://www.sap.com/platform/netweaver/components/aris/index.epx
9. *ДСТУ* 3918 – 1999 (ISO/IEC 12207:1995) Інформаційні технології. Процеси життєвого циклу програмного забезпечення.
10. *Советский* энциклопедический словарь / Гл.ред. А.М. Прохоров. – 3-е изд. – М.: Сов. Энциклопедия, 1984. – 1600 с.
11. *ISO/IEC* 15910:1999(E) Information technology.– Software user documentation process.
12. *ISO/IEC* 6592:2000(E) Information technology. – Guidelines for the documentation of computer-based application systems.
13. *Capability* Maturity Model Integration, Version 1.1 (CMMI-SW, V1.1), Carnegie Mellon Software Engineering Institute, 2002. – 645 p.
14. *Apple* Publications Style Guide – Apple Computer, Inc., 2006. –196 p.
15. *Якобсон А., Буч Г., Рамбо Дж.* Унифицированный процесс разработки программного обеспечения. – СПб.: Питер, 2002. – 496 с.
16. *Microsoft* Solutions Framework. Модель процессов MSF. Версия. 3.1. – Корпорация Майкрософт (Microsoft Corp.), 2002. – 44 с.
17. *Oracle Method*®. CDM Classic Method Handbook. Realease 2.6.0. – Oracle corp., 2000. – 248 p.
18. *Ambler, Scott W.* The Object Primer: Agile Model-Driven Development with UML 2.0.3 edition.– Cambridge University Press, 2004. – 572 p.
19. *ISO/IEC* TR 9294:1990(E). Information technology. – Guidelines for the management of software documentation.
20. *Бин Д.* XML для проектировщиков. Повторное использование и интеграция. – М.: КУДИЦ-ОБРАЗ, 2004. – 256 с.
21. *Сидоров М.О., Хоменко В.А., Авраменко О.А.* Реинженерия проектов программного обеспечения // Проблемы программирования. – 2006. – № 2–3. – С.31–38.
22. *Единая* система программной документации: сборник / Гос. стандарты СССР. – Изд. официальное. – М.: Изд–во стандартов, 1988. – 144 с.
23. *Basili V.R., Rombach H.D.* The TAME project: Toward improvement-oriented software environments // IEEE Transaction on Software Engineering. – 1988. – 14(6). – P. 758–773.