УДК 004.415.2

АВРАМЕНКО Е.А.

СРЕДСТВО РЕДОКУМЕНТИРОВАНИЯ НАСЛЕДУЕМОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В статье рассматривается метод редокументирования наследуемого программного обеспечения, ориентированный на технологию разработки; модель документа программного обеспечения. Предложен язык описания документов для редокументирования RDDL.

The article considers development technology oriented method of legacy software redocumentation; software document model It is proposed redocumentation document description language RDDL.

Введение

Редокументирование – это создание документации наследуемого программного обеспечения (ПО), отвечающей установленным требованиям или стандартам [1]. Задача редокументирования ПО возникла в 80-х годах прошлого столетия в связи с возросшим количеством наследуемого ПО, требующего реинженерии [1-3]. Такое ПО, чаще всего, не имеет документации или имеющаяся документация является устаревшей и неполной.

Существующие методы редокументирования ПО (например, структурное редокументирование [4], инкрементное редокументирование на основе Web-технологий [5]) направлены на решение задачи автоматизации отдельных аспектов документирования. Они также не учитывают требований к составу, структуре и содержанию документов.

В статье предлагаются средства для реализации метода редокументирования, применяемого при реинженерии наследуемого ПО. Средства позволяют повысить эффективность редокументирования путем автоматизированного создания документов, отвечающих требованиям технологии и фазы, выбранных для прямой инженерии. Рассмотрен язык описания документов Redocumentation Document Description Language (RDDL), построенный на основе модели документа ПО и обеспечивающий проведение редокументирования.

1. Метод редокументирования

Сущность метода редокументирования состоит в создании документации наследуемого ПО, состав, структура и содержание которой регламентируются требованиями технологии разработки, выбранной для прямой инженерии.

Состав документации при редокументировании определяется особенностями фазы, 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.1).

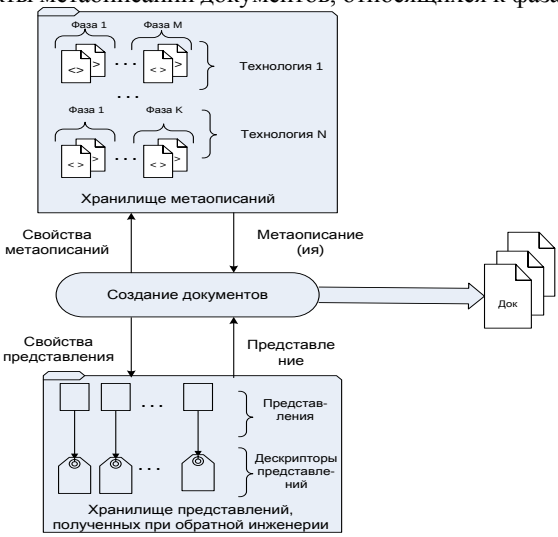

Рис.1. Схема использования метаописаний при редокументировании

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.

2. Модель документа ПО

Документ – это материальный объект, содержащий информацию в зафиксированном виде и специально предназначенный для её передачи во времени и пространстве [6]. 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 [7-9].

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

Структура документа S представлена множеством $S = \{se_i | i = 1..M\}$, где se_i – структурные элементы документации (СЭД). Множество всех СЭД документа $S = S_b \cup S_{in} \cup S_g$ можно рассматривать как объединение трех непересекающихся подмножеств:

- основных СЭД, определяемых информационным содержимым документа (например, главы, параграфы и пункты документа);

- идентификационных СЭД, предназначенных для идентификации документа (например, название и номер версии документа);
- общих СЭД, 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.

Общие и основные СЭД могут содержать вложенные элементы (СЭД), порядок расположения которых задается списком ребер. Например, для корневого СЭД дерева se_d список ребер имеет вид:

$$((se_d, se_1), (se_d, se_2), ..., (se_d, se_i), ..., (se_d, se_j), (se_d, se_n)) .$$

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

Между элементами множеств C и S определено такое соответствие R, что если задан некоторый ИЭД $ie \in C$, то и определен СЭД $se \in S$. 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 инкапсулированной информацией, отображающей часть ПО или ПО в целом в том или ином аспекте. Перечень требований, диаграмма вариантов использования, диаграмма компонентов, описание класса, диаграмма классов, изображение пользовательского интерфейса, схема БД являются примерами представлений ПО. Представление (v) имеет значение (z) (рисунок, текст, 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, которое описывает представление, например, примечание, подписи таблиц и рисунков.

Форма представления документа определяется правилами внешнего оформления, зависящими от формата носителя документа, которые обычо формируются путем установки значений для стандартизированных или корпоративных элементов стиля [10,11].

3. Язык метаописания лркументов

Для построения метаописаний документов предлагается специально разработанный язык RDDL, реализованный на основе XML [12]. Язык

RDDL базируется на рассмотренной модели документа и содержит средства для описания СЭД, ИЭД, представлений и пояснений.

Метаописание на языке RDDL состоит из элементов, принадлежащих следующим типам: корневому, идентификационному, общему и основному СЭД, ИЭД, представлению и пояснению, причем идентификационный и общий СЭД имеют подтипы. Элемент, принадлежащий корневому типу, в метаописании может быть только один, и по одному элементу каждого подтипа идентификационных и общих СЭД. Количество элементов, принадлежащих типам основного СЭД, ИЭД, представления и пояснения, в метаописании описании не ограничено.

Предлагаемый формальный язык характеризуется следующим:

- множествами меток и атрибутов;
- правилами, определяющими согласование и порядок следования элементов в метаописании;
- правилами описания элементов метаописания.

Множество меток определено для типов и подтипов элементов метаописания (табл. 1). Метка представляется идентификатором, обозначающим начало и конец описания элемента в метаописании.

Табл. 1

Метка	Подтип элемента метаописания	Тип элемента метаописания
document	–	корневой
projectName	название проекта	идентификационный СЭД
title	название документа	
version	номер версии документа	
revisionHistory	лист изменений	
tableofContents	содержание	общий СЭД
glossary	глоссарий	
introduction	введение	
listOfFigures	список иллюстраций	
listOfTables	список таблиц	
index	указатель	
bibliography	список литературы	
appendix	приложение	
item	–	основной СЭД
infoItem	–	ИЭД
view	–	представление
explanation	–	пояснение

Правила вложенности устанавливают элементы метаописания, имеющие вложенные элементы определенного типа (табл.2).

Табл. 2

Тип элемента метаописания	Тип вложенных элементов

корневой	идентификационный, общий и основной СЭД
общий СЭД	основной СЭД, ИЭД
основной СЭД	основной СЭД, ИЭД
ИЭД	представление, пояснение

Правило следования СЭД определяет порядок расположения вложенных СЭД в содержимом корневого элемента (рис.2). Порядок следования идентификационных СЭД устанавливается следующим образом: элементы, принадлежащие подтипам название проекта, название документа, номер версии документа и лист изменений. Порядок следования общих СЭД задается требованиями технологии. Порядок расположения основных СЭД друг относительно друга (следование и вложенность) устанавливается экспертом при создании метаописаний на языке RDDL.

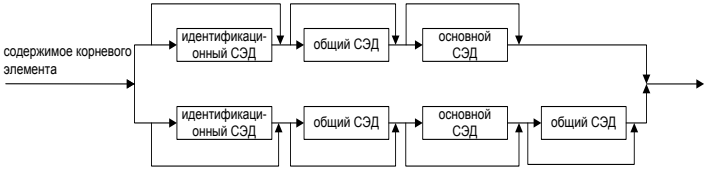

Рис.2. Правило следования СЭД в метаописании

Правил описания элементов каждого типа, базирующихся на структуре элемента метаописания (рис.3).

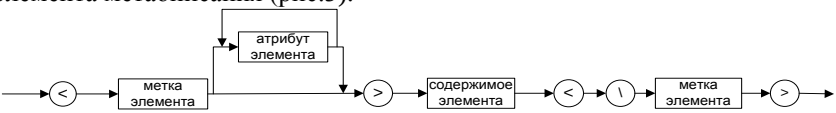

Рис.3. Структура элемента метаописания

Название метки выбирается из предопределенного в языке RDDL множества меток (табл.1), а содержимое элемента зависит от его вида, который может быть следующим:

- пустым – разрешены только атрибуты (содержимое элементов идентификационного СЭД и представления);
- вложенным – вложенные элементы и атрибуты (содержимое элементов корневого типа и типа ИЭД (рис.4));
- текстовым – текст (содержимое элемента, принадлежащего типу пояснение);
- смешанным – текст и вложенные элементы (содержимое элементов типов общий и основной СЭД (рис.5)).

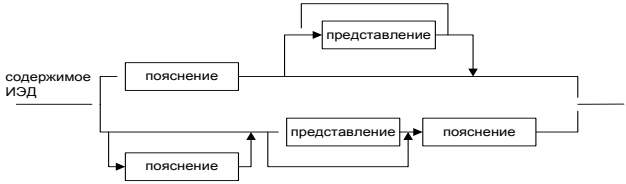

Рис.4. Описание содержимого ИЭД на языке RDDL

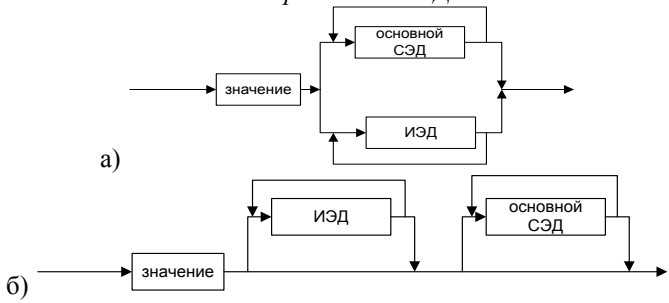

Рис.5. Описание содержимого СЭД на языке RDDL
а) для общего СЭД; б) для основного СЭД.

Описанные средства были применены при редокументирования одной из подсистем ПО, входящего в состав автоматизированной системы изготовления и учета документов государственного образца [13]. При реализации проекта были созданы метаописания документов фаз технологии RUP, которые использовались средствами автоматической генерации документов (рис.6).

```
<?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
                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.
                <item> 5.1 Overview
                        <infoItem>
                        <view type="ListOfPackages" notation="UML"></view>
                        <explanation>Package and Subsystem Layering <\explanation>
                        </infoItem>
                </item>
        </item>
        ...
<\document>
```

Рис. 6. Пример метаописания документа «Архитектура ПО» фазы Elaboration технологии RUP

Заключение

Редокументирование при реинженерии наследуемого ПО обеспечивает создание документации, которая используется в прямой инженерии. В методе редокументирования, ориентированном на технологию разработки, состав получаемой документации определяется фазой прямой инженерии, а структура и содержание документов – требованиями технологии. В качестве средства реализации метода предлагается использовать метаописания документов технологий, создаваемые на языке RDDL, разработанном на основе модели документа с помощью расширяемого языка разметки XML. Метаописания документов позволяют автоматизировать процесс редокументирования путем применения средств автоматической генерации документов ПО.

Список использованной литературы

1. Software Reengineering Assessment Handbook (SRAH). Version 3.0. JLC-HDBK-SRAH. March 1997.– 98p.
2. Chikofsky, Elliot J.; Cross II, James H. Reverse Engineering and Design Recovery: A Taxonomy. IEEE Software, January 1990.– pp.13-17
3. Tilley, Scott R. Perspective on Legacy System Reengineering (Draft Version 3), SEI Reengineering Center Publication, Summer 1996. – 124p.
4. Wong, Kenny; Tilley Scott R.; Müller, Hausi A. ; Storey, Margaret-Anne D. Structural Redocumentation: A Case Study. IEEE Software, Vol. 11, no. 6, January 1995. – pp. 501-520
5. Rajlish V. Incremental Redocumentation Using the Web. IEEE Software, September/October 2000. – pp.2-6
6. Советский энциклопедический словарь / Гл.ред. А.М.Прохоров.- 3-е изд. –М.: Сов. Энциклопедия, 1984. – 1600с.
7. ISO/IEC 15910:1999(E). Information technology – Software user documentation process
8. ISO/IEC 6592:2000(E). Information technology – Guidlines for the documentation of computer-based application systems
9. Software Engineering Institute, “Capability Maturity Model Integration, v1.1,” CMU/SEI-2002-TR- 002, ESC-TR-2002-002, December 2001
10. ISO/IEC FDIS 18019:2003(E). Software and systems engineering — Guidelines for the design and preparation of user documentation for application software
11. Microsoft® Manual of Style for Technical Publications, Third Edition: Microsoft Corporation Editorial Style Board, 2004 – 381 p.
12. Бин Д. XML для проектировщиков. Повторное использование и интеграция. – М.:КУДИЦ-ОБРАЗ, 2004. – 256с.
13. Сидоров М.О., Хоменко В.А., Авраменко Е.А. „Реинженерия проектов программного обеспечения”. Проблемы программирования. – НАНУ. – 2,3, 2006. – с.31-38