

К. Ю. Романовский

РАЗРАБОТКА ПОВТОРНО-ИСПОЛЬЗУЕМОЙ ДОКУМЕНТАЦИИ СЕМЕЙСТВА ТЕЛЕФОННЫХ СТАНЦИЙ СРЕДСТВАМИ ТЕХНОЛОГИИ DOCLINE

Введение. Промышленное программное обеспечение (ПО) обычно сопровождается значительным пакетом документов, описывающих различные аспекты его работы – от первого знакомства до сложного администрирования. С увеличением сложности ПО растет и сложность документации. Особенно явственно это проявляется для активно развивающихся семейств программных продуктов (наборов сходных программных систем, разрабатываемых в рамках единого технологического процесса, на основе общих повторно используемых активов [1]). В данном случае необходимо разрабатывать и сопровождать пакеты документов для каждого из продуктов, такие документы имеют много пересечений, но вместе с тем и множество отличий.

Для разработчиков ПО существует множество подходов организации повторного использования кода – от процедур в алгоритмических языках до широкомасштабных методов совместного создания семейств программных продуктов. В разработке документации активно применяется ряд известных XML-технологий – DITA [2], DocBook [3] и др., но повторное использование в них реализовано достаточно «прямолинейно»: либо фрагмент текста можно подключить «как есть», без изменений (или с изменениями, выражающимися в терминах условных блоков), либо требуется вручную делать копию этого фрагмента и редактировать уже копию (cut and paste). Вместе с тем даже такой простой и привычный механизм повторного использования программного кода, как процедура, допускает широкие возможности настройки с помощью параметров.

Для поддержки повторного использования документации был предложен метод DocLine [4, 5], обеспечивающий адаптивное повторное использование документации семейств программных продуктов, планирование которого реализуется с помощью техник визуального моделирования, в частности, диаграмм возможностей (Feature Diagrams) [6]. Адаптивность дает возможность сконфигурировать общие активы независимо для каждого контекста использования.

DocLine является XML-технологией, и каждый программист способен научиться эффективно пользоваться ею. Но документацию разрабатывают не программисты, а технические писатели, так что вопрос о том, насколько легко они смогут применять DocLine, является чрезвычайно актуальным^{*)}. Кроме того, интересен вопрос о том,

Романовский Константин Юрьевич – старший преподаватель кафедры системного программирования математико-механического факультета Санкт-Петербургского государственного университета. Количество опубликованных работ: 11. Научные направления: повторное использование в разработке ПО, разработка семейств программных продуктов. E-mail: kromanovsky@yandex.ru.

^{*)}Во многих случаях именно необходимость применять XML/HTML оказывается тем водоразделом, по которому проходит непреодолимая граница компетентности – люди специальностей, не связанных с компьютерами, нередко отказываются от таких средств работы с текстами. Так одним из основных свойств систем управления контентом для разработки Интернет-сайтов, способствующим их продвижению на рынке, является предоставление контент-менеджерам удобного интерфейса, скрывающего работу с HTML (подробнее о системах управления контентом см. [7]).

©К.Ю. Романовский, 2009

как в DocLine будет выглядеть реальная промышленная документация, сколь много в ней будет повторов, какие из конструкций языка окажутся задействованы, а какие нет. Для ответа на эти вопросы была проведена апробация DocLine на базе семейства телекоммуникационных систем «Квант-Е» компании ЗАО «Ланит-Терком». С помощью DocLine был выполнен перенос двух руководств пользователя для двух разных продуктов семейства (общий объем около 300 с.) из формата Adobe FrameMaker в DocLine, выделены и формализованы повторно используемые фрагменты. В данной работе представлены и проанализированы полученные результаты.

1. Обзор существующих подходов. DocLine находится на стыке двух предметных областей: методов разработки семейств программных продуктов и технологий и подходов разработки технической документации.

Семейство программных продуктов (product line) – это набор продуктов, совместно продвигаемых на рынке и разрабатываемых на основе общих активов с помощью хорошо определенного метода [1]. В качестве повторно используемых активов выступают программные компоненты, модели анализа и проектирования, практики менеджмента, тесты и пр. [1]. Существует ряд подходов, эффективно поддерживающих повторное использование общих активов. Особо отметим метод адаптивного повторного использования Пола Бассета [8, 9] и диаграммы возможностей, предложенные в работах [6, 10]. Метод Бассета [8, 9], а также язык XVCL [11], предложенный позже группой ученых из университета Сингапура, обеспечивают языково-независимое управление повторным использованием программного кода. При этом общие активы могут быть адаптированы к особенностям различных контекстов использования с помощью специализированного препроцессора. Как оригинальный подход Бассета, так и язык XVCL не применяются для разработки документации, поскольку нуждаются в серьезных модификациях для адаптации к этой задаче.

Основное назначение диаграмм возможностей, получивших широкое распространение в различных методах разработки семейств программных продуктов, – формализация и визуальное отображение общих и частных свойств продуктов семейства. Диаграммы возможностей в оригинальном варианте могут применяться только на этапах анализа и, в меньшей степени, проектирования и никак не связаны с артефактами последующих этапов, такими как программный код и документация.

Из технологий разработки технической документации отметим DocBook [3] и DITA [2]. DocBook обеспечивает разделение содержания документа и его форматирования, что позволяет создать единое исходное представление документа (single source [12]) и на его основе автоматически генерировать документацию в разных целевых форматах (PDF для печати, HTML Help для контекстной справочной системы и т. п.). Повторное использование в классическом смысле (с явным выделением общих активов и последующим их подключением в точки вхождения) в DocBook практически не развито.

В технологии DITA предлагается идея повторного использования крупных и логически завершенных фрагментов текста (топиков) в различных контекстах. DITA позволяет также организовать повторное использование небольших фрагментов текста – отдельных слов, фраз, терминов, фрагментов предложений и т. д. При этом, в DITA отсутствует механизм параметризации топиков, а значит возможности конфигурирования общих активов существенно ограничены.

Итак, существуют методы разработки семейств программных продуктов, которые хорошо поддерживают планируемое повторное использование, однако они не адаптированы для разработки документации. Вместе с тем, существующие языки и средства

разработки документации не поддерживают специфичные для семейств программных продуктов возможности повторного использования – в различных подходах к разработке документации реализуются лишь отдельные идеи, такие как единое исходное представление в DocBook.

2. Обзор DocLine. DocLine восполняет разрыв между методами разработки семейств программных продуктов и подходами разработки технической документации. Фактически, DocLine основывается на лучших решениях из обеих областей и интегрирует их в единую технологию, ориентированную на разработку документации семейств программных продуктов и поддерживающую весь процесс разработки – от проектирования документов до их публикации.

- DocLine развивает идеи метода Бассета [9] для случая документации, а также добавляет специфические именно для документации конструкции повторного использования.
- Также мы применяем диаграммы возможностей [6] для описания схемы повторного использования фрагментов документации. При этом диаграммы не являются самостоятельными артефактами, а реализуют высокоуровневое представление документации в целом и в любой момент разработки синхронизированы с документацией.
- Для поддержки форматирования текста и генерации документов в различных форматах DocLine использует DocBook [3] как язык разметки.

DocLine ориентирован на документацию семейств программных продуктов, которая может содержать массу повторяющихся фрагментов и при этом активно эволюционировать. Благодаря DocLine все изменения повторно-используемых компонент будут автоматически распространяться по всем контекстам их включения.

DocLine состоит из языка, модели процесса разработки документации и пакета программных средств. Язык DRL (Documentation Reuse Language – язык повторного использования документации) предназначен для планирования и реализации повторно используемой документации [4]. Для планирования DRL предлагает визуальную нотацию DRL/GR, включающую три вида диаграмм – главные диаграммы, диаграммы вариативности и продукта. Для реализации адаптивности и собственно написания текстов DRL предлагает XML-нотацию DRL/PR. Сам DRL не содержит средств форматирования текста – для решения этой задачи он интегрирован с популярной технологией DocBook [3] (еще один эффект такой интеграции – для технических писателей, знакомых с DocBook, переход на применение DRL не составляет большого труда).

DocLine предлагает две модели процесса разработки документации, распространенных в мире семейств программных продуктов: проактивную [13] – от подробного анализа и создания общих активов к разработке отдельных документов, и реактивную [14] – от разработки одного документа к выделению общих активов и выпуску последующих документов. На практике чаще встречается легковесный процесс разработки. Основной приоритет в реактивном процессе – реализация определенной бизнес-цели, в данном случае создание документации конкретного продукта. Когда возникает необходимость разработать документацию новых продуктов семейства, технический писатель анализирует схожие черты и различия продуктов семейства, выявляет возможности повторного использования, создает общие активы на основе документации первого продукта. Затем создается документация новых продуктов на основе общих активов.

Инструментальный пакет DocLine интегрирован в среду разработки Eclipse и содержит графический редактор схем повторного использования, XML-редактор текстов

документации, средства проверки корректности текстов и средства генерации конечных документов в различных целевых форматах (PDF, HTML и т. п.).

2.1. Обзор языка DRL.

Планирование. Для структуризации документации в DRL применяется понятие *информационный продукт* – фактически, это шаблон документа, такого как руководство пользователя или справочник. Для каждого продукта семейства, в котором востребован тот или иной информационный продукт (например, контекстная справка) необходимо создать *финальный информационный продукт*, представляющий собой сконфигурованный вариант информационного продукта. Далее на основе финального информационного продукта автоматически получается конечный документ, поставляемый в составе документации. Строительные блоки, из которых составляются информационные продукты, – это *информационные элементы* – контекстно-независимые повторно-используемые фрагменты текста. Каждый информационный элемент или продукт может включать другие информационные элементы. Эти включения могут быть обязательными или не обязательными, а также их можно объединять в группы двух видов: OR-группы, допускающие включение любого набора элементов из группы (такие группы используются для перечисления «опций», которые могут в любом сочетании войти в разные продукты, например способы набора номера: тональный, импульсный, голосовой), и XOR-группы, допускающие включение не более чем одного элемента (применяются для объединения взаимоисключающих вариантов, например кнопочный или дисковый телефонный аппарат).

Планирование осуществляется с помощью DRL/GR. На рис. 1–3 приведены примеры всех видов диаграмм. Далее они будут подробно обсуждаться.

Адаптивность. Основной механизм адаптивного повторного использования, реализованный в DRL, – конфигурирование информационных элементов. Для того чтобы была возможность сконфигурировать информационный элемент под особенности конкретного контекста использования, нужно предусмотреть в нем точки расширения в тех местах, в которых возможна вариативность. Далее, в процессе конфигурирования информационного элемента, в эти точки расширения вносится соответствующий дополнительный текст (с точками расширения допустимы следующие операции: удаление содержимого, добавление текста перед или после точки расширения). Набор таких изменений оформляется в виде специальной конструкции DRL/PR – *адаптера*.

Поясним идею конфигурируемых информационных элементов на примере – рассмотрим документацию семейства телефонных аппаратов с определителем номера вызывающего абонента. В ней может содержаться информационный элемент «Получение входящего вызова», содержащий такой текст:

После получения входящего вызова телефон запрашивает информацию о звонящем абоненте и отображает ее на экране. (1)

Но различные типы телефонов могут иметь разные опции индикации вызывающего абонента, например телефон для слабовидящих может вместо визуального отображения проговаривать номер абонента. Тогда пример (1) выглядел бы так:

После получения входящего вызова телефон запрашивает информацию о звонящем абоненте и проговаривает ее. (2)

Для обеспечения преобразования примера (1) к виду (2) с помощью техники адаптивного повторного использования информационный элемент следует определить следующим образом (в синтаксисе DRL):

```
<infelement id=CallerIdent> (3)
После получения входящего вызова телефон запрашивает информацию о звонящем абоненте <nest id=DisplayOptions> и отображает ее на экране</nest>.
</infelement>
```

В этом примере мы определяем информационный элемент (тег <infelement/>) и точку расширения (тег <nest/>). Когда информационный элемент включается в определенный контекст, каждая точка расширения может быть удалена или дополнена специфичным текстом без необходимости изменения исходного информационного элемента. Если не задано никаких расширений, информационный элемент из примера (3) будет преобразован в текст примера (1). Следующие настройки преобразуют его в вид примера (2):

```
<infelemref infelemid=CallerIdent> (4)
<replace-nest nestid=DisplayOptions> и проговаривает ее</replace-nest>
</infelemref>
```

В этом примере содержатся ссылка на информационный элемент (<infelemref/>), определенный в примере (3), и замена текста точки расширения новым текстом (<replace-nest/>).

Для упорядочивания терминологии в документах используются словари. *Словарь* – это конструкция DRL/PR, которая включает в себя набор пар (имя, значение). В произвольной точке документа можно подставить значение элемента словаря, указав его имя. В словаре могут быть такие элементы как название продукта, версия, набор поддерживаемых операционных систем и т. п. Также DRL предлагает обобщение словарей – каталоги, которые предназначены для унификации представления в тексте однородных элементов, более сложных, чем термины. *Каталог* – это набор элементов-кортежей (имя, атрибут 1, атрибут 2, ...). Так, каталог команд пользовательского интерфейса может содержать коллекцию команд интерфейса, имеющих имя, пиктограмму, описание, «горячую клавишу», текст всплывающей подсказки, список побочных эффектов, правила применения и т. д.

3. Семейство АТС «Квант-Е». Для апробации было выбрано промышленное семейство телекоммуникационных систем, выпускаемых компанией ЗАО «Ланит-Терком». Базовый продукт семейства – программное обеспечение электронной автоматической телефонной станции типа «Квант-Е» (далее – *станция* или *АТС*). На основе базового продукта разработан ряд модификаций, включая сельские, офисные, городские, транзитные телефонные станции, а также специализированные станции (например, с поддержкой IP-телефонии). К настоящему моменту в телефонных сетях России установлены сотни экземпляров представленного семейства^{*)}.

Мы выбрали два продукта семейства – базовая городская телефонная станция (далее – *городская станция* или *ГАТС*), а также станция специального назначения (*специальная станция* или *САТС*). Апробация проводилась на примере руководства пользователя для одной из подсистем рассматриваемого семейства – рабочего места оператора станции (*РМО*), обеспечивающего поддержку технического обслуживания и эксплуатации станции.

САТС является, фактически, «вырезкой» из ГАТС, которая была доработана под индивидуальные требования заказчика. Относительно документации заказчик выдвинул требование, чтобы в его комплекте были описаны только те функции, которые во-

^{*)}Более подробно о телефонных станциях «Квант-Е» см. [15].

шли в его комплект поставки, а все остальные, общие, были из описания убраны. Так, из описания РМО всего семейства станций родился документ, описывающий только РМО САТС. Именно к двум этим документам и была применена технология DocLine.

4. Ход апробации. Основной задачей апробации было выполнить перенос существующей документации семейства телефонных станций «Квант-Е», созданной в Adobe FrameMaker, в DocLine с организацией повторного использования документации. Как следует из описания семейства, в этой документации должны быть повторы текста.

В апробации участвовали технический писатель и консультант по семейству телефонных станций. До начала апробации технический писатель не был знаком ни с пакетом DocLine, ни с технологией DocBook, ни с XML, ни с исходным семейством. Руководил процессом один из разработчиков технологии DocLine (автор этой статьи), который также разрешал технические вопросы, связанные с эксплуатацией программного продукта DocLine.

В ходе апробации выполнялись следующие работы: изучение и анализ предметной области; планирование повторного использования; выделение и спецификация переиспользуемых компонент; задание форматирования документов средствами DocBook.

Необходимо отметить, что это скорее виды работ, чем этапы процесса, поскольку работа велась итеративно, т. е. в разные моменты апробации происходил возврат к тем или иным работам. Например, изучение и анализ предметной области производились в течение всего проекта, по мере необходимости. Тем не менее отчасти их можно считать и этапами, поскольку их порядок отражает основное распределение работ в проекте*).

В целом трудозатраты на проект оказались следующими: 20 дней работы технического писателя, 2 дня работы консультанта по семейству телефонных станций, 2 дня работы автора DocLine.

Приступим к рассмотрению отдельных работ, выполненных в ходе апробации.

4.1. Изучение и анализ предметной области. Эта часть проекта оказалась очень важной по следующим причинам. Технический писатель, организующий повторное использование, должен хорошо разбираться в функциональности описываемых продуктов – существенно лучше, чем при написании обычных документов. В последнем случае технические писатели часто подходят к написанию документации формально, не пытаясь анализировать детально специфику продуктов, а описывают отдельные окна приложений и их свойства. Такую же склонность продемонстрировал и наш технический писатель, и потребовалось немало сил, чтобы убедить его в необходимости детально изучить описываемые продукты. Поиск похожих и почти похожих фрагментов текста можно производить синтаксически, но гораздо эффективнее понимать, чем продукты отличаются, и подходить к изменениям в документах от изменений в функциональности. В целом изучение и анализ предметной области занял 3 дня работы технического писателя и 1 день работы консультанта по семейству телефонных станций.

4.2. Планирование повторного использования. Основным инструментом здесь был DRL/GR. Сначала технический писатель с помощью диаграмм начал изображать общую структуру документов – главы, разделы, подразделы т. д. и хотел уже на этой картине отмечать повторно используемые компоненты. Однако такой подход был отброшен, поскольку диаграммы получались большие и необозримые. После того, как на диаграмме вариативности рис. 2 были представлены разделы первого уровня, дальнейшая декомпозиция проводилась только для тех разделов и подразделов, внутри

*)О взаимоотношении видов работ и этапов в процессах по разработки ПО см. [16].

которых присутствовала вариативность. Результаты этого этапа – диаграммы рис. 1–3. Таким образом, мы выяснили варианты крупноблочного повторного использования (т. е. какие разделы присутствуют/отсутствуют в документации для разных продуктов), а также в каких разделах есть незначительные отличия. Данная работа заняла 5 дней работы технического писателя, 1 день работы консультанта по семейству телефонных станций и половину дня работы специалиста по DocLine.

4.3. Выделение и спецификация переиспользуемых компонент. Для разделов документации, в которых для ГАТС и САТС имелись отличия (такие разделы были найдены при планировании повторного использования) средствами DRL/PR была проведена настройка адаптивности. Они были выделены в информационные элементы, в них вставлялись необходимые точки расширения, создавались специализированные информационные элементы – те, которые входили в конечные документы. Эта деятельность была тесно связана с анализом и изучением предметной области и оказалась самой трудоемкой, заняв 8 дней работы технического писателя и 1 день работы специалиста по DocBook. Работа была существенно облегчена благодаря средствам рефакторинга, имеющимся в DocLine, автоматизировавшим такие операции как выделение информационного элемента, извлечение текста в точку расширения, выделение элемента в словарь и т. п.

4.4. Задание форматирования документов средствами DocBook. Требовалось, чтобы внешний вид документации после применения DocLine ничем не отличался от исходного. Потому нужно было задать необходимые свойства форматирования средствами DocBook и настроить подходящим образом утилиты публикации DocBook. Это потребовало половину дня работы специалиста по DocLine, так как выяснилось, что тонкая настройка DocBook достаточно сложна для технического писателя, не имеющего ранее опыта работы с XML-технологиями. Технический писатель затратил на данную часть работы 4 дня.

5. Особенности использования DocLine. Теперь посмотрим на апробацию с другой точки зрения, а именно, рассмотрим, какой оказалась вариативность текстов и как удалось ее выразить средствами DocLine.

5.1. Вариативность продуктов семейства. Вариативность текстов прямо следует из вариативности функциональности продуктов, которые описываются этими текстами. Как уже говорилось выше, исторически САТС разрабатывалась как ограниченная версия ГАТС и уже в процессе эволюции появились разнообразные отличия в функциях, реализованных в обеих системах. Таким образом, в нашем случае можно выделить базовый продукт и его документацию (ГАТС) и продукт-потомок (САТС). Рассмотрим следующие виды вариаций функциональности (и документации), характерные для такого подхода.

- *Удаление* – из базовой версии системы удаляется функциональность, которая не требуется пользователям нового продукта. Соответственно описание этой функциональности удаляется из документации. Это может быть отдельный фрагмент текста, целая глава или набор глав. К сожалению, нередко текст, который требуется удалить, распределен по различным разделам документации – где-то удаляется глава, где-то отдельные абзацы, элементы списка или строчки таблиц и т. п. Так происходит потому, что описание удаляемой функциональности часто не локализовано в тексте, а распределено по разным его местам.
- *Добавление* – в базовую версию добавляется новая функциональность и ее описание должно быть внесено в пользовательскую документацию. Как и в случае с

Таблица 1. Частота изменений в документации

Характер изменений	Частота, %
Удаление	32
Добавление	0
Изменение	5
Локальное изменение	26
«Параллельные места»	37

удалением, изменения текста могут затрагивать целый ряд разделов документации. Эта операция не характерна для нашего случая, так как в САТС не произошло добавления каких-либо существенных новых функциональных возможностей по сравнению с ГАТС.

- *Изменение* – значительно меняется какое-либо функциональное свойство, что для документации выражается в требовании описать измененные атрибуты этого свойства в виде блока подряд идущего текста либо изменения рассредоточены в разных фрагментах документа.
- *Локальное изменение* – меняется лишь отдельный атрибут функционального свойства, например количество обслуживаемых абонентов. Для пользовательской документации такое изменение функциональности превратится в небольшое локальное изменение текста. Например, при изменении количества обслуживаемых абонентов будет изменено число в сводной таблице функциональных возможностей.

Есть также особый вид изменений, не затрагивающий функциональность – это «*параллельные места*»: одна и та же функциональность описывается в разных документах различными словами. «Параллельные места», в целях унификации документов, нужно по возможности убирать, выбирая единый способ описания одной и той же информации.

Еще раз подчеркнем, что удаление, добавление и изменение особенны тем, что некоторая информация (описание одной функциональной возможности) не локализована в тексте, поэтому ее добавление или удаление требует особой внимательности. Ошибки здесь могут привести к потере корректности текста.

В табл. 1 приведены процентные соотношения указанных типов изменений в документации РМО ГАТС и САТС.

5.2. Схема вариативности. В результате планирования повторного использования в рассматриваемой документации были созданы модели с помощью DRL/GR (см. рис. 1–3).

На рис. 1 представлена главная диаграмма, на которой отображены продукты семейства (левая секция) и единственный имеющийся у нас информационный продукт. На рис. 2 приведена диаграмма вариативности для информационного продукта «Руководство пользователя» – общее дерево всех разделов для обоих документов. Корневой узел диаграммы – информационный продукт «Руководство пользователя», вариативность которого формализуется (прямоугольник с двойной рамкой). Узлы диаграммы – информационные элементы (прямоугольники с обычной рамкой). Связь между узлами (обозначается линией) означает включение. Полный круг на одном из концов линии означает, что соответствующее включение не обязательно (может в одни продукты входить, а в другие нет, как «Описание реализации»). Группы включений информационных элементов обозначаются путем объединения нескольких узлов в групповой узел



Рис. 1. Главная диаграмма семейства АТС

(полый круг характеризует OR-группу, а сплошной – XOR-группу). Повторно используемые разделы документации отмечены жирными прямоугольниками, а те «листовые» разделы (т. е. не содержащие других подразделов), которые входят в оба документа, но отличаются отдельными деталями, включают указания на то, сколько в них имеется точек расширения (т. р.).

Проработку диаграммы вариативности «в глубину» целесообразно ограничить и отображать только информационные элементы, которые варьируются крупноблочно (т. е. входят или нет целиком) или содержат внутри себя мелкие отличия для двух документов. Остальные разделы (см., например, многие разделы первого уровня на рис. 2) также выносятся в информационные элементы, но далее не раскрываются в DRL в иерархию, хотя размечаются средствами DocBook с целью задания надлежащего форматирования. Таким образом, «листьями» на диаграммах вариативности являются разделы, которые идентичны во всех контекстах применения, либо разделы, которые варьируются, но не включают других разделов – вообще, либо варьирующихся.

В документации раздел «Главное окно» почти идентичен для обоих типов станции, но содержит несколько фрагментов, которые различаются лишь порядком расположения. Для описания такой вариативности было решено выделить «перемещающиеся» фрагменты в информационные элементы и объединить в XOR-группы – каждая такая группа описывает вхождения одного фрагмента в различные точки объемлющего блока.

На рис. 3 представлена диаграмма продукта для РМО САТС. Корневой узел – финальный информационный продукт «Руководство пользователя РМО САТС». Узлы диаграммы – информационные элементы, которые фактически входят в руководство пользователя для САТС. Единственный допустимый вид связей между узлами – безусловное включение. Таким образом, эта диаграмма является «вырезкой» для САТС из общей модели документации (модели вариативности). Для разделов, имеющих точки расширения, указано, сколько из них задействовано. Например, запись 1/7 для раздела «Панель состояния тракта» на рис. 3 говорит о том, что из семи точек расширения в случае документации для САТС задействована одна.

5.3. Задание адаптивности.

В руководствах пользователя РМО ГАТС и САТС есть несколько общих подразделов («Абонентская панель», «Карта обслуживания абонентских линий», «Соединительные линии»), которые содержат ряд отличий, не разрешаемых на уровне включения

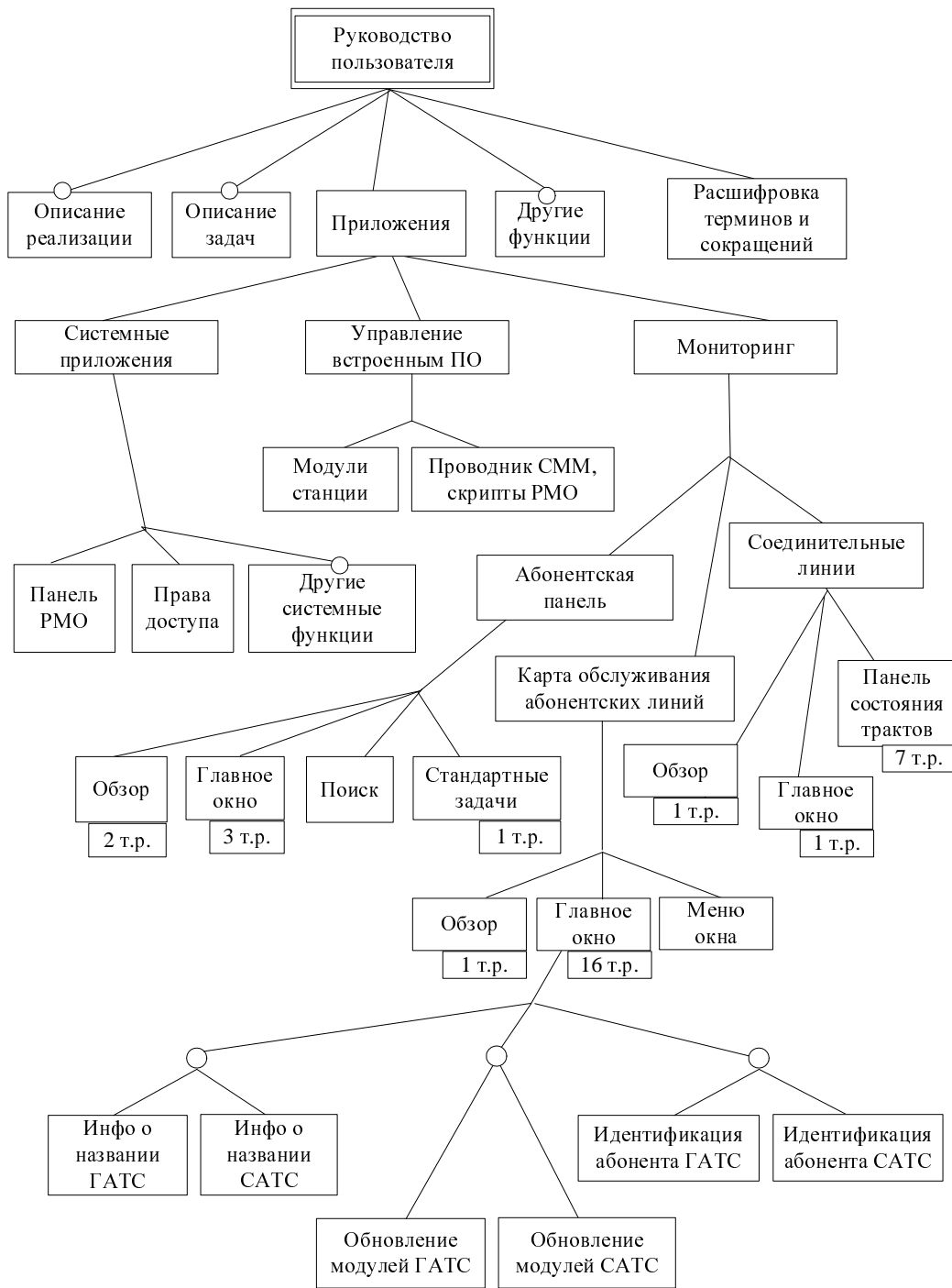


Рис. 2. Структура документации семейства АТС

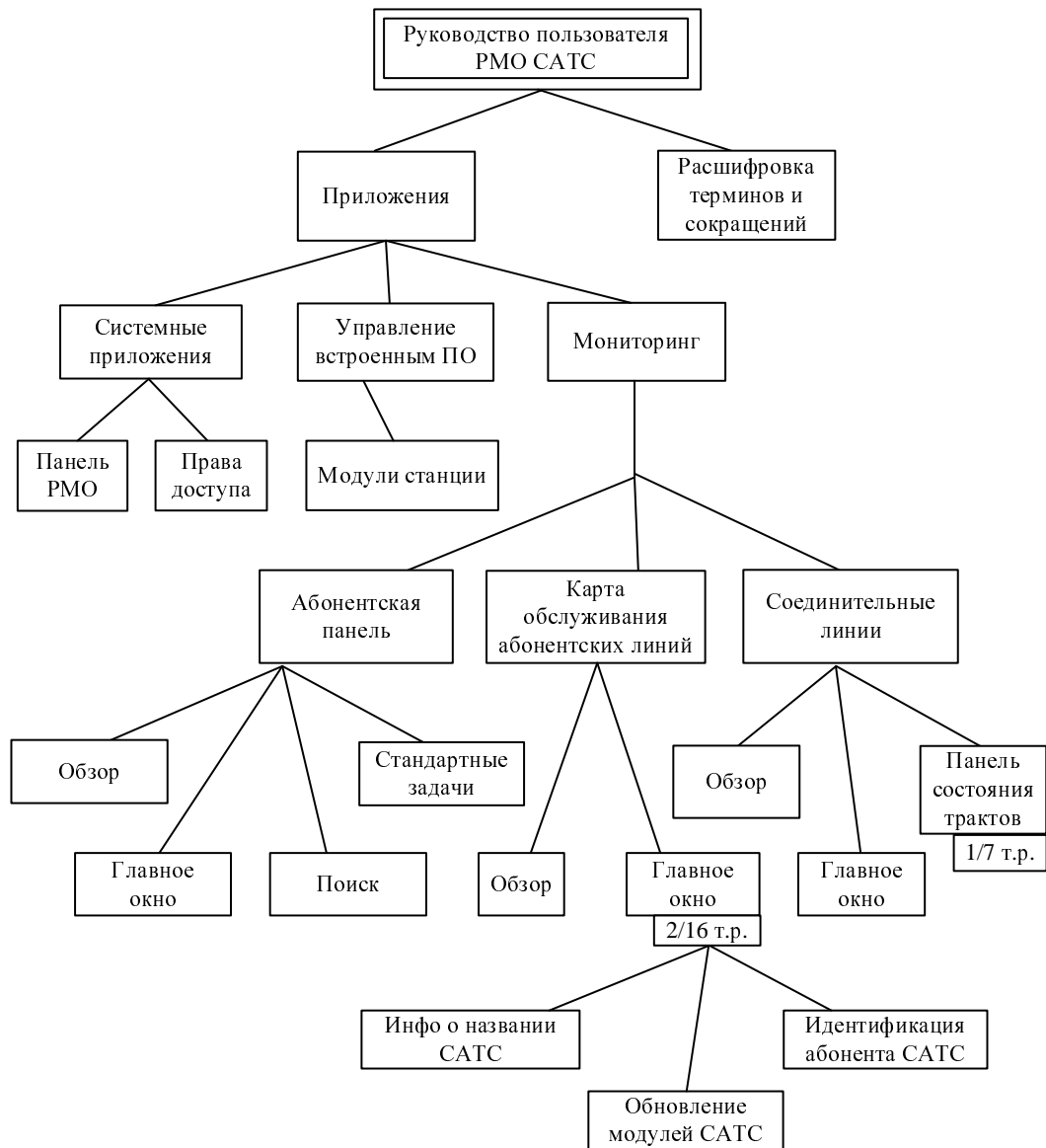


Рис. 3. Диаграмма документации продукта РМО САТС

Таблица 2. Пример локальных изменений в тексте

РМО ГАТС	РМО САТС
Чтобы активизировать автоматическое обновление состояний абонентских линий, необходимо нажать на правую часть этой кнопки и в появившемся меню выбрать необходимый период времени обновления (из представленных: 5, 10 или 30 с). При этом будут обновляться состояния абонентов, которые <i>заказаны в списке опрашиваемых модулей</i>	Чтобы активизировать автоматическое обновление состояний абонентских линий, необходимо нажать на правую часть этой кнопки и в появившемся меню выбрать необходимый период времени обновления (из представленных: 1 , 5, 10 или 30 с). При этом будут обновляться состояния <i>только тех</i> абонентов, которые <i>были доступны при последней загрузке данных из станции</i>

или исключения целых информационных элементов. В DocLine для формализации таких отличий используются точки расширения и механизм адаптации.

В табл. 2 показан фрагмент документации из раздела «Карта обслуживания абонентских линий», описывающий управление автоматическим обновлением состояний абонентских линий. Содержательная разница между реализацией этой функции для САТС и ГАТС состоит в том, что для САТС необходимо обеспечивать более частые обновления, при этом в ГАТС обновляется состояние только явно запрошенных абонентских модулей, а в САТС – всех модулей, которые были доступны при последнем обмене данными со станцией. Различающиеся детали выделены полужирным курсивом. Фактически, фрагменты очень похожи, но не идентичны. В исходной документации версия для САТС была получена копированием и последующим исправлением фрагмента для ГАТС. Рассмотрим, как такие «точечные» изменения можно описать с помощью DRL. Фрагмент текста, описывающий автоматическое обновление, будет выглядеть следующим образом:

```
<Infelement id=AutoUpdate>
  Чтобы активизировать автоматическое обновление состояний абонентских линий,
  необходимо нажать на правую часть этой кнопки и в появившемся меню выбрать
  необходимый период времени обновления (из представленных:
  <Nest id=Period>5, 10 или 30 с</Nest>). При этом будут
  обновляться состояния <Nest id=Condition/> абонентов, которые
  <Nest id=Constraint>заказаны в списке опрашиваемых модулей</Nest>.
</Infelement>
```

Все позиции, где допустимы изменения, отмечены точками расширения (<nest/>).

В документацию ГАТС указанный элемент будет включен «как есть», а для САТС требуется описать ряд изменений:

```
<Adapter infelemrefid=AutoUpdate_ref>
  <Insert-Before nestid=Period>1, </Insert-Before>
  <Insert-Before nestid=Condition>только тех</Insert-Before>
  <Replace-Nest nestid=Constraint>были доступны при последней загрузке
  данных из станции</d:Replace-Nest>
</Adapter>
```

Изменения описываются с помощью конструкции <Adapter/>, в которой указываются точки расширения, требующие модификации и, собственно, сами модификации – в данном случае добавление данных о периоде обновления, и изменение описания множества обновляемых модулей.

Как видно из примера, DRL позволяет подключать в нескольких документах похожие, но не идентичные фрагменты текста – это и есть реализация адаптивности повторного использования (т. е. возможности адаптироваться под требования контекста). В данном случае с помощью механизма адаптации DocLine формализовано изменение вида «локальное изменение».

6. Анализ и дискуссия. Первый позитивный результат апробации заключается в том, что в реальной документации продуктов семейства действительно много повторов, которые варьируются в деталях (см. вышеприведенные примеры). Таким образом, адаптивное повторное использование документации действительно оправдано и механизмы адаптивности (точки расширения и адаптеры) сыграли существенную роль в успехе апробации – без них в большинстве случаев повторное использование было бы невозможно, даже в рассмотренной ситуации, когда один из продуктов семейства был изначально построен как «вырезка» из базового продукта.

Относительно применимости языка DRL по результатам данного пилотного проекта можно сказать следующее. DRL/GR оказался востребован, особенно диаграммы вариативности. Именно с их помощью происходило планирование повторного использования, а также обсуждение его результатов между техническим писателем, специалистом по DocLine и консультантом по семейству телефонных станций. Это особенно важно на первых порах, пока технический писатель еще не овладел до конца приемами практического применения DocBook. Вся дальнейшая организация повторного использования происходит на основе диаграмм вариативности. Диаграммы продуктов были полезны, хотя и в меньшей степени, главная диаграмма оказалась небольшой и малосодержательной.

Из DRL/PR активно применялись информационные элементы, точки расширения и адаптеры. Для унификации терминологии были задействованы словари. Каталоги практически не потребовались, поскольку в документации не было массово встречающихся однородных объектов. Мы намерены расширить DRL по результатам апробации для удобной формализации фрагментов текста, входящих в несколько контекстов, но в различных точках.

Таким образом, использование DocLine предполагает высокие требования к квалификации технических писателей – необходимо, чтобы у них были способности к системному мышлению и широкий кругозор в рамках данного проекта для того, чтобы они могли организовывать повторное использование, не только основываясь на синтаксическом подобии отдельных фрагментов текста, но и исходя из свойств описываемого ПО. Требуются также ресурсы для организации и осуществления дополнительных коммуникаций между техническими писателями и другими участниками проекта (менеджерами, разработчиками, архитекторами) для эффективной передачи знаний о продуктах семейства – об их схожих свойствах и различиях.

По результатам апробации был сделан вывод о том, что в DocLine должны быть добавлены средства управления планированием повторного использования на основе модели возможностей (feature model [6]) семейства продуктов. Необходимо также добавить средства поддержки автоматизации синтаксического поиска похожих фрагментов текстов, а также отличий (последнее было бы очень полезно в нашем проекте).

Программные средства DocLine показали себя достаточно удобными для эффективной разработки документации. Основные сложности возникали с редактированием текста на XML и настройками DocBook – в следующих версиях программного пакета планируется упростить такие работы.

В целом выяснилось, что применение DocLine при наличии модели вариативности практически ничем не отличается по трудоемкости от применения, например, DocBook, однако при этом DocLine добавляет средства планирования и механизм адаптивного повторного использования. Основные предположения об эффективности DocLine относятся к возможности применения подобных XML-технологий при разработке технической документации в принципе. Скорее всего, прорыв в России произойдет при достижении российским рынком ПО определенной зрелости: когда процессы производства станут более стабильными, устоявшимися, когда будет много семейств программных продуктов, развивающихся в разных сегментах рынка не по одному году. Ведь пользуются же в российских книжных издательствах для верстки книг продуктом TeX, который намного сложнее, чем DocLine. В то же время, в западных университетах, во-первых, специально готовят технических писателей (специальность «технические коммуникации»), а во-вторых, учат их XML-технологиям. Такие XML-технологии как DocBook и DITA активно применяются в индустрии. А до тех пор, пока проекты относительно небольшие, семейства программных продуктов несложные, нет высоких требований к качеству документации, говорить о DocLine рано. Ведь затраты на внедрение нашей технологии оправданы, если не просто стоит задача разработки двух-трех похожих документов, а когда таких документов больше и, главное, есть процесс их эволюции, занимающий существенное время (от года и больше) – вносятся исправления и доработки, а также дополнения. Вот здесь-то выясняется, что, имея такую структуру повторного использования, изменения достаточно внести в одно место – во все остальные они распространятся автоматически.

Заключение. Итак, данный пилотный проект был нацелен на «статическую» апробацию DocLine – требовалось посмотреть, как в языке DRL выражаются повторно используемые фрагменты реально существующей промышленной документации, удобны ли средства планирования повторного использования и механизм адаптивности, удобно ли работать в инструментальном пакете DocLine. В целом результаты получились обнадеживающие – полученные в DocLine документы внешне ничем не отличались от разработанных в Adobe FrameMaker, затраты на перенос были не слишком велики. Документация была унифицирована (убраны «параллельные места», унифицирована терминология). Также в результате апробации были обозначены новые направления развития DocLine: совершенствование языка DRL, разработка средств семантического управления повторным использованием на основе функциональных возможностей продуктов семейства, автоматический синтаксический поиск похожих фрагментов текста, поиск отличий.

«Динамические» аспекты эффективности DocLine – соотношение затрат и выгод от применения технологии в рамках интенсивно эволюционирующего семейства программных продуктов – остались пока в стороне и будут исследованы в дальнейшем.

Литература

1. *Clements P., Northrop L.* Software Product Lines: Practices and Patterns. Boston, MA: Addison-Wesley, 2002. 608 p.
2. *Day D., Priestley M., Schell D.* Introduction to the Darwin Information Typing Architecture – Toward portable technical information // <http://www-106.ibm.com/developerworks/xml/library/x-dita1/>.
3. *Walsh N., Muellner L.* DocBook: The Definitive Guide. New York: O'Reilly, 1999. 644 p.
4. *Романовский К. Ю., Кознов Д. В.* Язык DRL для проектирования и разработки документации семейств программных продуктов // Вестн. С.-Петерб. ун-та. Сер. 10: Прикладная математика, информатика, процессы управления. 2007. Вып. 4. С. 110–122.
5. *Кознов Д. В., Романовский К. Ю.* DocLine: метод разработки документации семейств программных продуктов // Программирование. 2008. № 4. С. 1–13.
6. *Czarnecki K., Eisenecker U.* Generative Programming: Methods, Tools, and Applications. Reading, Mass.: Addison Wesley Longman, 2000. 864 p.
7. *Кознов Д. В., Остапенко О.В.* Разработка Интернет-приложений в небольшой компании с применением product line-подхода // Системное программирование. Вып. 2 / Под ред. А. Н. Терехова, Д. Ю. Булычева. СПб.: Изд-во С.-Петерб. ун-та, 2006. С. 169–190.
8. *Bassett P.* Frame-Based Software Engineering // IEEE Software. 1987. Vol. 4, N 4. P. 9–16.
9. *Bassett P.* Framing software reuse – lessons from real world. Upper Saddle River, NJ: Yourdon Press, 1996. 365 p.
10. *Kang K., Cohen S., Hess J., Novak J. et al.* Feature-Oriented Domain Analysis (FODA) Feasibility Study: Techn. report, CMU/SEI-90-TR-21. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1990. 87 p.
11. *Jarzabek S., Bassett P., Hongyu Zhang, Weishan Zhang.* XVCL: XML-based variant configuration language // Proc. of 25th Intern. Conf. on Software Engineering. 2003. P. 810–811.
12. *Rockley A., Kostur P., Manning S.* Managing Enterprise Content: A Unified Content Strategy. Berkeley, CA: New Riders Press, 2002. 592 p.
13. *Clements P.* Being Proactive Pays Off. IEEE Software July/August 2002. P. 28–29.
14. *Krueger C.* Eliminating the Adoption Barrier. IEEE Software July/August 2002. P. 30–31
15. *Кияев В. И., Кищенко Д. М., Окомин И. С.* Опыт усовершенствования и стандартизации процесса создания ПО цифровых телефонных станций // Системное программирование. Вып. 2 / Под ред. А. Н. Терехова, Д. Ю. Булычева. СПб.: Изд-во С.-Петерб. ун-та, 2006. С. 219–239.
16. *Кознов Д.В.* Языки визуального моделирования. Проектирование и визуализация программного обеспечения. Учеб. пособие. СПб.: Изд-во С.-Петерб. ун-та, 2004. 171 с.

Статья рекомендована к печати проф. А. Н. Тереховым.

Статья принята к печати 25 декабря 2008 г.

Summary

Romanovsky K. Y. Introducing reuse to phone exchanges family documentation by means of DocLine technology.

The task of developing software product lines becomes actual due to increasing demands for creation of complex software systems. The main advantage that can be achieved by joint developing a set of software-intensive systems is the reuse of various common assets including source code. Multi-purpose technical documentation is a necessary part of modern software, but the existing methods of developing software product lines do not pay proper attention to the task of creating the documentation. An XML-based method DocLine for developing technical documentation of software product lines is proposed. This method is intended for reusing text fragments. DocLine uses visual modeling technique to support the design of the documentation. Also DocLine provides a means for adapting reusable text fragments according to the needs of various usage contexts. An adoption of DocLine is presented. Two manuals for a phone exchanges family software (about 300 pages in total) originally developed in Adobe FrameMaker were re-authored in DocLine with extraction and specification of reusable parts. Adoption results and classification of products variability types recognizable from the perspective of documentation development are presented. Conclusions are also given regarding competence of a technical writer who is capable of effective use of the proposed technology in industrial settings.

Key words: software product lines, reuse, technical documentation, visual modeling

Романовский К. Ю. Разработка повторно-используемой документации семейства телефонных станций средствами технологии DocLine.

Задача разработки семейств программных продуктов приобретает актуальность в условиях растущих требований к созданию сложного программного обеспечения. Основное преимущество, которое можно получить совместно разрабатывая ряд программных систем – повторное использование различных активов, в том числе программного кода. Всесторонняя техническая документация – необходимая составляющая современного программного обеспечения, однако в методах разработки семейств программных продуктов задаче разработки документации не уделяется должного внимания. Автор предлагает XML-метод разработки технической документации семейств программных продуктов DocLine, ориентированный на повторное использование фрагментов текста. Для проектирования документации DocLine предлагает средства визуального моделирования. Также DocLine поддерживает механизмы адаптации повторно-используемого текста к особенностям контекста использования.

В данной работе представлена апробация DocLine. Был выполнен перенос двух документов для программного обеспечения семейства телефонных станций общим объемом около 300 страниц из формата Adobe FrameMaker в DocLine с выделением и спецификацией повторяющихся частей. В работе анализируются результаты апробации, классифицируются варианты вариативности продуктов семейства, значимые с точки зрения задачи разработки технической документации. Также делаются выводы о компетенциях технического писателя, который мог бы эффективно использовать подобную технологию в своей работе.

Ключевые слова: повторное использование, семейства программных продуктов, техническая документация, визуальное моделирование

УДК 004.434:004.42

Романовский К. Ю. **Разработка повторно-используемой документации семейства телефонных станций средствами технологии DocLine** // Вестн. С.-Петерб. ун-та. Сер. 10. 2009. Вып. 2. С. 00–00

Представлен проект реализации повторно-используемой документации семейства телефонных станций с помощью технологии DocLine. В ходе проекта на базе разрозненной документации двух программных продуктов создана повторно-используемая документация. Проанализированы виды отличий документации нескольких сходных продуктов. Библиогр. 16 назв. Ил. 3. Табл. 2.

Ключевые слова: повторное использование, семейства программных продуктов, техническая документация, визуальное моделирование