

Содержание:

ВВЕДЕНИЕ

Тема курсовой работы «Языки гипертекстовой разметки».

Достаточно часто программная документация имеет специфические особенности. Например, используются специальные для данной предметной области шаблоны описания, или есть существенные ограничения технологического процесса подготовки документации.

В данной работе рассматриваются различные системы документирования программного обеспечения, и делается выбор соответствующего инструментария. Также делается вывод о необходимости доработки готовой системы документирования и описываются разработанные для конкретных случаев технология и программные средства, которые позволяют решить поставленную задачу более экономичным образом.

Постановка задачи: Обосновать актуальность использования XML документов как универсального формата для передачи данных.

За пример обоснования взят достаточно большой по объему комплекс программного обеспечения входит подсистема «Быстрых отчетов», построенных на основе SQL- запросов. Подсистема должна иметь программную документацию. Количество отчетов - достаточно большое, несколько сотен, готовить и, тем более, сопровождать описание отчетов традиционными методами достаточно сложно и дорого.

Надо выбрать наиболее подходящий инструментарий для документирования программной подсистемы и, в случае необходимости, внести необходимые доработки в технологию и программные средства. Составление программной документации - очень важная составляющая разработки программного продукта [1].

Поскольку материал, из которого состоит техническая документация, - текст на естественном языке, применить к нему модульный подход оказывается сложнее, чем к исходному коду. В тексте документации достаточно особенностей,

обусловленных лексическими и грамматическими особенностями, традициями, сложившимися мнениями стилистики и эстетики.

Принцип единого источника и технологии, позволяют применить модульный принцип к документированию [3]. Благодаря этому принципу мы можем несмотря на все своеобразие текста параметризовать почти одинаковое и формировать из этих блоков документы в том виде, в котором их предпочитает получать аудитория.

Понятие представления данных и обоснование проблемы

Данные в компьютере представляются в виде кода, который состоит из единиц и нулей в разной последовательности.

Код - набор условных обозначений для представления информации. Кодирование - процесс представления информации в виде кода.

Кодирование текстовой информации.

Кодовая таблица - это внутреннее представление символов в компьютере. Во всем мире в качестве стандарта принята таблица ASCII (American Standard Code for Information Interchange - Американский стандартный код для обмена информацией). Для хранения двоичного кода одного символа выделен 1 байт = 8 бит. Учитывая, что каждый бит принимает значение 0 или 1, количество их возможных сочетаний в байте равно $2^8 = 256$. Значит, с помощью 1 байта можно получить 256 разных двоичных кодовых комбинаций и отобразить с их помощью 256 различных символов. Эти комбинации и составляют таблицу ASCII.

Для сокращения записей и удобства пользования кодами символов используют шестнадцатеричную систему счисления, состоящую из 16 символов - 10 цифр и 6 латинских букв: A,B,C,D,E,F. Так, буква S будет представлена в шестнадцатеричной системе по сравнению с двоичной более компактным кодом 53. Стандарт ASCII определяет первые 128 символов от 0 до 127: цифры, буквы латинского алфавита, управляющие символы. Первые 32 символа являются управляющими и предназначены в основном для передачи команд управления. Вторая половина

таблицы от 128 до 255 – национальный алфавит[3].

Кодирование чисел.

Двоичная система счисления обладает такими же свойствами, что и десятичная, только для представления чисел используется не 10 цифр, а всего две. Соответственно и разряд числа называют не десятичным, а двоичным.

Для кодирования числа, участвующего в вычислениях, используется специальная система правил перевода из десятичной системы исчисления в двоичную. В результате число будет записано двоичным кодом, т.е. представлено различным сочетанием всего двух цифр - 0 и 1.

Кодирование графической информации.

Создавать и хранить графические объекты в компьютере можно двумя способами - как растровое изображение или как векторное изображение. Для каждого типа изображения используется свой способ кодирования.

Векторное изображение представляет собой графический объект, состоящий из элементарных отрезков и дуг. Положение этих элементарных объектов определяется координатами точек и длиной радиуса. Для каждой линии указывается ее тип (сплошная, пунктирная, штрих - пунктирная), толщина и цвет. Информация о векторном изображении кодируется как обычная буквенно-цифровая и обрабатывается специальными программами.

Кодирование звуковой информации.

Звуковая информация может быть представлена последовательностью элементарных звуков (фонем) и пауз между ними. Каждый звук кодируется и хранится в памяти. Вывод звуков из компьютера осуществляется синтезатором речи, который считывает из памяти хранящийся код звука[3].

Самой большой проблемой в представлении данных является разнообразие ПО с помощью которого эти данные кодируются в память ПК и отображаются пользователю. Для удобства работы необходимо представление данных, которое

будет отображаться на любом устройстве без предварительной установки ПО.

Применение технологии единого источника требует подготовки. Сначала проектируется структура единого источника, разрабатываются шаблоны и стили оформления, устанавливается, и настраивается инструментарий для формирования документов. Эта стадия отнимает время в начале проекта и предъявляет достаточно высокие требования к квалификации разработчика технической документации. Но потом начинается рутинная работа по написанию текста и его загрузке в единый источник.

В любой момент на основе введенного текста можно сформировать документы, с большей или меньшей степенью готовности. На этой стадии мы получаем отдачу от сделанных вложений. Раздел, рисунок, таблица, абзац, любой другой фрагмент, который должен появляться в нескольких местах, при необходимости достаточно исправить однократно в едином источнике. Замечательно, что это касается не только текста, но и структурных решений.

Однократное внесение изменений в шаблон, допустим, руководства пользователя, достаточно, чтобы все документы этого типа после очередной автоматической обработки были изменены нужным образом. Принцип единого источника в документировании, как и модульный принцип в программировании, помогает организовать работу коллектива, распределив между ними более или менее изолированные подзадачи.

Таким образом, единственный источник - это не только техническое, но еще и организационное решение.

Языки гипертекстовой разметки. Обзор существующих технологий представления данных.

В 1989 году Тим Бернерс-Ли предложил руководству международного центра высоких энергий (CERN) проект распределенной гипертекстовой системы, которую он назвал World Wide Web (WWW), Всемирная паутина.

Первоначально идея системы состояла в том, чтобы при помощи гипертекстовой навигационной системы объединить все множество информационных ресурсов CERN в единую информационную систему. Технология оказалась настолько удачной, что дала толчок к развитию одной из самых популярных в мире глобальных информационных систем. Практически в сознании большинства пользователей глобальной компьютерной сети Internet сама эта сеть ассоциируется

с тремя основными информационными технологиями:

- электронная почта (e-mail);
- файловые архивы FTP;
- World Wide Web.

Успех технологии World Wide Web определен двумя основными факторами: простотой и использованием протоколов межсетевого обмена семейства TCP/IP, (Transmission Control Protocol, протокол управления передачей/Internet Protocol, протокол Internet), которые являются основой Internet. Практически все пользователи Сети одновременно получили возможность попробовать себя в качестве создателей и читателей информационных материалов, опубликованных во Всемирной паутине. Но и популярность самого Internet во многом вызвана появлением World Wide Web, так как это первая сетевая технология, которая предоставила пользователю простой современный интерфейс для доступа к разнообразным сетевым ресурсам. Простота и удобство применения привели к росту числа пользователей WWW и привлекли внимание коммерческих структур.

Далее процесс роста числа пользователей стал лавинообразным, и так продолжается до сих пор. При этом сама технология на начальном этапе была чрезвычайно проста. Дело в том, что при разработке различных компонентов технологии (языка гипертекстовой разметки HTML (HyperText Markup Language, язык разметки гипертекста), протокола обмена гипертекстовой информацией HTTP, спецификации разработки прикладного программного обеспечения CGI и др.) предполагалось, что квалификация авторов информационных ресурсов и их оснащенность средствами вычислительной техники будут минимальными.

HTML, HyperText Markup Language - "язык разметки гипертекста", стандартный язык разметки документов во Всемирной паутине (Web, WWW). Является объектно-ориентированным декларативным языком, имеющим строгую объектную иерархию (DOM-модель, Document Object Model). Большинство веб-страниц создаются при помощи языка HTML (или XHTML). Язык HTML интерпретируется браузерами и отображается в виде документа, в удобной для человека форме. HTML является приложением ("частным случаем") SGML (стандартного обобщенного языка разметки) и соответствует международному стандарту ISO 8879. Расширенный HTML (eXtensible HTML, XHTML) является приложением XML.

Язык HTML был разработан британским учёным Тимом Бернерсом-Ли приблизительно в 1989-1991 годах в стенах Европейского совета по ядерным исследованиям в Женеве (Швейцария). HTML создавался как язык для обмена научной и технической документацией, пригодный для использования людьми, не являющимися специалистами в области вёрстки. HTML успешно справлялся с проблемой сложности SGML путём определения небольшого набора структурных и семантических элементов - дескрипторов. Дескрипторы также часто называют "тегами". С помощью HTML можно легко создать относительно простой, но красиво оформленный документ. Помимо упрощения структуры документа, в HTML внесена поддержка гипертекста. Мультимедийные возможности были добавлены позже.

HTML - теговый язык разметки документов. Любой документ на языке HTML представляет собой набор элементов, причём начало и конец каждого элемента обозначается специальными пометками - тегами. Элементы могут быть пустыми, то есть не содержащими никакого текста и других данных (например, тег перевода строки `
`). В этом случае обычно не указывается закрывающий тег. Кроме того, элементы могут иметь атрибуты, определяющие какие-либо их свойства (например, размер шрифта для элемента `font`). Атрибуты указываются в открывающем теге.

Изначально язык HTML был задуман и создан как средство структурирования и форматирования документов без их привязки к средствам воспроизведения (отображения). В идеале, текст с разметкой HTML должен был без стилистических и структурных искажений воспроизводиться на оборудовании с различной технической оснащённостью (цветной экран современного компьютера, монохромный экран органайзера, ограниченный по размерам экран мобильного телефона или устройства и программы голосового воспроизведения текстов). Однако современное применение HTML очень далеко от его изначальной задачи. Например, тег `<TABLE>`, несколько раз использованный для форматирования страницы, которую вы на данный момент читаете, предназначен для создания в документах самых обычных таблиц, но, как можно убедиться, здесь нет ни одной таблицы. С течением времени, основная идея платформонезависимости языка HTML была отдана в своеобразную жертву современным потребностям в мультимедийном и графическом оформлении.

Текстовые документы, содержащие разметку на языке HTML (такие документы традиционно имеют расширение `.html` или `.htm`), обрабатываются специальными приложениями, которые отображают документ в его форматированном виде. Такие приложения, называемые "браузерами" или "интернет-обозревателями", обычно

предоставляют пользователю удобный интерфейс для запроса веб-страниц, их просмотра (и вывода на иные внешние устройства) и, при необходимости, отправки введённых пользователем данных на сервер. Наиболее популярными на сегодняшний день браузерами являются Internet Explorer, Mozilla Firefox, Apple Safari, Google Chrome и Opera.

Версии HTML

HTML 1.0 - множество нестандартных языков, существовавших в сети до 1995 г.

HTML 2.0, стандарт RFC 1866, одобренный 22 сентября 1995 года;

HTML 3.0 была предложена Консорциумом всемирной паутины (W3C) в марте 1995 года и обеспечивала много новых возможностей, таких как создание таблиц, "обтекание" изображений текстом и отображение сложных математических формул. Даже при том, что этот стандарт был совместим со второй версией, реализация его была сложна для браузеров того времени.

HTML 3.1 официально никогда не предлагалась, и следующей версией стандарта HTML стала 3.2, в которой были опущены многие нововведения версии 3.0, но добавлены нестандартные элементы, поддерживаемые браузерами Netscape Navigator и Mosaic.

HTML 3.2 - введена 14 января 1997 года;

HTML 4.0 - введена 18 декабря 1997 года. В версии HTML 4.0 произошла некоторая "очистка" стандарта. Многие элементы были отмечены как устаревшие и не рекомендованные (англ. deprecated). В частности, элемент font, используемый для изменения свойств шрифта, был помечен как устаревший (вместо него рекомендуется использовать таблицы стилей CSS).

HTML 4.01 (изменения, причём более значительные, чем кажется на первый взгляд) - 24 декабря 1999 года;

ISO/IEC 15445:2000 (так называемый ISO HTML, основан на HTML 4.01 Strict) - 15 мая 2000 года.

HTML 5 - в настоящее время находится в разработке. Черновой вариант спецификации языка появился в Интернете 20 ноября 2007 года. Конец разработки

запланирован на 2014 год.

XHTML 1.0. В 1998 году консорциум Всемирной паутины начал работу над новым языком разметки, основанном на HTML 4, но соответствующим синтаксису XML. Впоследствии новый язык получил название XHTML. Первая версия XHTML 1.0 одобрена в качестве Рекомендации консорциума Всемирной паутины 26 января 2000 года.

XHTML 2.0. Планируемая версия XHTML 2.0 должна была разорвать совместимость со старыми версиями HTML и XHTML, но 2 июля 2009 года консорциум Всемирной паутины объявил, что полномочия рабочей группы XHTML2 истекают в конце 2009 года. Таким образом, была приостановлена вся дальнейшая разработка стандарта XHTML 2.0

Сообщество WHATWG (англ. Web Hypertext Application Technology Working Group), начиная с 2004 года, разрабатывается спецификация Web Applications 1.0, часто неофициально называемая "HTML 5", которая расширяет HTML (впрочем, имея и совместимый с XHTML 1.0 XML-синтаксис) для лучшего представления семантики различных типичных страниц, например форумов, сайтов аукционов, поисковых систем, онлайн-магазинов и т. д., которые не очень удачно вписываются в модель XHTML 2.

В зависимости от целей и объемов документации, готовящимся разработчиками создаются различные технологические цепочки с использованием следующих наиболее популярных программных продуктов [4 - 6]:

□ Microsoft Word - PDF Текст пишется с использованием текстового редактора Microsoft Word и затем конвертируется в формат PDF. технологическая цепочка позволяет готовить бумажные публикации, а также комплекты электронной документации в формате PDF. Используется в тех случаях, когда требования к издательскому качеству документации невысокие. Преимуществами являются легкость внедрения и возможность рецензирования и редактирования материалов всеми, кто имеет к ним доступ. Недостатком является невысокое качество верстки.

□ Adobe FrameMaker - PDF текст верстается с использованием программного комплекса Adobe FrameMaker и затем конвертируется в формат PDF. Технологическая цепочка позволяет готовить бумажные публикации, а также комплекты электронной документации в формате PDF. Возможно также преобразования в другие форматы (RTF, HTML, различные приложения XML и т. д.). Используется для подготовки публикаций профессионального издательского

качества.

К недостаткам относится необходимость затрат на внедрение и невозможность изменения материалов «на лету».

□ Технологии подготовки Контекстной справки Технологические цепочки этого типа используются для создания контекстной справки в одном из распространенных форматов: WinHelp, HTML Help и т.д. Цепочка может включать специализированную программную систему (ForeHelp, eHelp, RoboHelp) или обходиться без нее. Технологии подготовки контекстной справки требуют затрат на внедрение и согласования действий между техническим верстальщиком и командой разработчиков.

□ Технология единого источника на основе DocBook Текст технической документации сначала пишется на языке разметки DocBook и является единственным источником. Две параллельные технологические цепочки позволяют получить файл в формате PDF (который в дальнейшем распространяется в электронном виде или служит основой для бумажной публикации) или файл в формате HTML Help. Технология единого источника на основе DocBook является оптимальным решением в тех случаях, когда необходимо составить и подать техническую документацию как в форме бумажных (или электронных) документов, так и в форме системы контекстной справки. При этом все изменения вносятся в единый источник, выходные же файлы любого формата при необходимости генерируются из единого источника. Таким образом, их содержание всегда соответствует актуальному состоянию исходного текста, написанного на DocBook. Технология требует значительных затрат труда на внедрение.

□ технология DITA Основной постулат технологии DITA состоит в том, что текст технической документации состоит из фрагментов (так называемых топиков), типизированных, причем каждому типу отвечает определенный план описания. При этом технология DITA позволяет автору декларировать новые типы Топиков для нужд конкретного проекта и задавать для каждого типа план описания. В комплекте инструментария DITA уже есть готовые стили, реализующие все наиболее важные функции формирования документов примерно на том же уровне, на котором это делают стили DocBook. Технология DITA позволяет, с одной стороны, создавать новые типы топик, а с другой, использовать для их обработки уже имеющийся стиль. Это достигается благодаря приему, который в объектно-ориентированном программировании известен как наследование. Технология DITA является более сложной в сравнении с DocBook и требует еще больших затрат

на внедрение.

Из всех рассмотренных технологий наиболее приемлемой для начального внедрения принципа единого источника кажется технология DocBook. Она является в достаточной степени гибкой и мощной и одновременно не такой сложной, как DITA. К тому же DocBook является открытой системой (Open Source проект). Технология DocBook / XML основой технологической платформы DocBook / XML служит одноименная проблемно-ориентированный язык разметки [7]. Он предназначена для записи текста технической документации на программы, алгоритмические языки, компьютерное оборудование и другие решения в области информационных технологий, чем принципиально отличается от большинства форматов хранения текстовых данных (но не XML-языков).

Представление данных в виде XML-документов.

Следует отметить, что для описания схемы XML-файлов уже сейчас существует с десяток форматов. Однако стандартными из них являются лишь два: DTD (старый формат, являющийся частью XML 1.0) и XML Schema (утвержден в мае 2001 года). Далее под XML-схемой будет подразумеваться файл в формате XML Schema (.xsd).

Чем хорош стандарт XML-схема?

1. Он предоставляет значительную часть информации, которая хранится обычно в схеме базы данных (реляционной, иерархической и т.д.).
2. В файл со схемой можно внести дополнительную информацию, так что стандартные валидаторы будут ее игнорировать, а специализированные программы будут ее использовать.

В качестве модели данных в данный формат заложена модифицированная модель объект/отношение (сущность/связь, Entity/Relationship в оригинальном варианте). Особенности приложения модели к данной задаче [1] следующие:

1. Объекты:

1. Нет различия между **слабыми** и **сильными**.
2. Тип объектов обязательно имеет имя.
3. Объекты могут иметь **метаданные** (не путать со свойствами - см. далее).

2. Свойства (реквизиты):

1. Только **простые**, нет **составных** (структур).
 2. Есть **ключевые свойства**, уникальные в контексте отношения.
 3. Возможны **однозначные** и **многозначные** свойства. Под многозначными свойствами понимается неупорядоченное множество попарно различных элементов (т.е. порядок элементов не сохраняется).
 4. Свойства могут **отсутствовать** у экземпляров.
 5. Нет **производных** свойств (таких, как сумма чего-нибудь).
 6. Свойства (как и объекты) могут иметь **метаданные**.
3. **Отношения**. Возможны отношения как со **степенью** два (бинарные), так и более.
4. **Подтипы** отсутствуют.

По сути единственным серьезным дополнением модели сущность/связь является концепция метаданных.

Метаданные документа - это некоторая дополнительная информация, которая семантически не может быть отнесена к свойствам документа. Например, идентификатор документа является внутренней информацией, которая актуальна для хранилища, но ее бессмысленно делать свойством.

Метаданные свойства - это дополнительная информация о свойстве, которая отражает его представление в прикладной программе и влияет на его обработку. Например, прямоугольник привязки в системах распознавания текстов или разметка реквизита в информационных системах являются метаданными свойствами.

Метаданные могут быть только простых типов или являться отношениями - образующими (иными словами, являться ссылками на другие типы).

Документ - это множество объектов, связанных отношениями, с одним выделенным объектом - корневым.

Коллекция документов - это множество, элементами которого являются **Документы** и **Коллекции документов**.

Данный формат предназначен для организации передачи данных между различными подсистемами. Поэтому программы разбора будут иметь содержательную функциональность, и, как следствие, не самую высокую производительность. Следовательно, не рекомендуется применять полновесный разборщик в "mission-critical" приложениях, где от этого может пострадать общая производительность системы

В языке DocBook / XML предвиденные средства описания фрагментов, присущих технической документации, например, обозначения клавиш, названия переменных, сроки, различные врезки (взвешивание, подсказки, предупреждения), листинги, описания выполняемых пользователем процедур. разметка, что задается языке DocBook / XML, носит преимущественно функциональный характер: автор указывает роль, которую тот или другой фрагмент играет в тексте, а не способ его внешнего оформления. Такой подход сковывает автора, зато позволяет добиться явной независимости содержания и оформления выходного документа и унифицировать некоторые важные качества стиля изложения при работе нескольких авторов в одном проекте.

Для набора текста входящего документа могут применяться различные программы, от обычного «Блокнота» до развитых XML-редакторов. Поскольку требования к формату входящего документа определяются спецификой используемых языков разметки, выбор редактора перестает быть важным вопросом, который необходимо решить на уровне проекта.

Каждый автор может работать в том редакторе, который ему удобен. По отношению к заданию набора текста в формате DocBook / XML редакторы, существуют сегодня, можно разделить на следующие группы:

- Текстовые редакторы;
- XML-редакторы с интерфейсом текстового процессора;
- XML-редакторы, упрощающие набор разметки.

Требования к оформлению выходных документов в различных проектах различны, поэтому попытки выпустить исчерпывающий набор XSLT-стилей, обречен на неудачу.

Создавать XSLT-стиле заново в каждом проекте тоже нельзя, потому что сроки и стоимость этой работы выйдут неприемлемыми. На практике обычно используют свободно распространяемый комплект стандартных стилей DocBook XSL [7]. Он поддерживает основные элементы макетов и целевые форматы, а также хорошо приспособлен к доработке для нужд конкретного проекта.

Благодаря архитектуре XSLT адаптация стандартных стилей не требует модификации их кода. Новые правила, располагаются в отдельных файлах, дополняют или замещают стандартные. Выполнены доработки привязаны к

конкретной копии стандартных стилей, следовательно:

- Доработка не конфликтуют друг с другом, и ими легко обмениваться;
- Возможность обновления версий DocBook XSL ограничена только их обратной совместимостью.

DocBook успешно используется при разработке и поддержке технической документации, особенно когда речь идет об объемах в тысячи и десятки тысяч страниц. Технология позволяет решить ряд типичных задач, возникающих при разработке технической документации, таких как:

- Разработка и оформление документации в соответствии с отраслевыми и государственными стандартам (ЕСКД, ЕСПД, КСАС)
- Автоматизация процедур нормоконтроля документации;
- Подготовка документации в различных электронных форматах, предназначенных для печати и просмотра на экране монитора: PDF, HTML, HTML Help, Java Help, Unix Man Pages;
- Поддержка многопользовательских режимов подготовки документации.

Технология DocBook прекрасно подходит предприятиям, чья продукция сопровождается технической документацией, подготовленной и оформленной в соответствии с требованиями ГОСТ 2, 19 и 34 серии (ЕСКД, ЕСПД, КСАС).

Однако, даже такая развитая технология, как DocBook, требует некоторой доработки при конкретном ее применении. Подготовка специализированной программной документации

Одной из подсистем которую можно рассмотреть для примера, есть подсистема «Быстрые отчеты» SQL, которая служит жить для выдачи данных о таблицах-справочниках системы на основании SQL запросов. Подсистема «Быстрые отчеты» SQL должна иметь полноценную программную документацию, которая включает разделы руководства пользователя и руководства программиста (см. рис. 1).

В состав программной документации также входит экранная помощь пользователю отчетов. Подготовка программной документации для подсистемы «Быстрые отчеты» SQL имеет ряд важных особенностей:

- SQL запросы могут достаточно часто меняться во время эксплуатации системы темы, вместе с этим также изменяется внешний вид отчетов, и, следовательно, должна меняться документация;
- Программная документация должна формироваться во всех указанных выше видах - Для конечного пользователя, для программиста, экранная помощь;
- Документация для пользователя должна быть представлена в нескольких форматах - в формате набора HTML-страниц и формате PDF-файла;
- Программная документация в виде экранной помощи реализуется с помощью специальной программной системы и поэтому исходная информация для нее имеет быть представлена в виде XML - файлов специального формата;
- При эксплуатации системы могут появляться новые отчеты SQL и удаляться существующие.

Рис. 1. Диаграмма использования документации подсистемы

Все эти факторы приводят к тому, что система подготовки программной документации, а особенно - система сопровождения программной документации должны быть очень динамичными и должны позволять быструю подготовку различных форматов документации из одного источника.

В качестве основной системы документирования, на основании проведенного анализа, выбранная система документирования DocBook (реализация для Windows).

Для получения необходимой документации экранной помощи в виде XML-файла, структура которого определена внешней системой подготовки экранной помощи, необходимо разработать скрипт конвертации средствами XSLT (речь преобразования XML - документов). Схема подготовки документации для подсистемы «Быстрые отчеты» SQL будет иметь вид, изображенный на рис. 2. Однако, подготовка описаний в формате DocBook для большого числа отчетов будет весьма трудоемкой, по этому проще готовить начальные описания по значительно упрощенному шаблону «XML в специальном формате» (для системы помощи).

Модифицированная схема подготовки документации приведена на рис. 3. Такое изменение направления потока данных в рабочем процессе приведет к экономии многих часов работы операторов по подготовке исходных описаний отчетов.

Рис. 2. Схема подготовки документации (вариант 1)

Такой процесс подготовки документации является эквивалентным хорошо известному в программировании процессу компиляции и сборки большого программного продукта.

Рис. 3. Схема подготовки документации (вариант 2)

ЗАКЛЮЧЕНИЕ

XML-схема является, чрезвычайно мощным, стандартом представления документации. Но эта мощь влечет за собой сложность программного обеспечения, работающего со схемой.

В процессе исследования рассмотрены различные системы документирования программного за непосредственно печенье и обоснован выбор технологии DocBook. Приведенный пример доработкой готовой системы документирования в конкретном случае задача подготовки специализированной программной документации (системы выдачи программных отчетов) решен более экономичным образом чем с использованием иных программных продуктов.

Список литературы

1. Липаев В.В. Документирование и управление конфигурацией программных средств / В.В. Липаев. - М.: Синтег, 2008. - 220 с.
2. Липаев В.В. Качество программных средств / В.В. Липаев. - М: Янус-К, 2012. 400 с.
3. Технология DocBook и принцип единого источника [Электронный ресурс]. Режим доступа к ресурса: <http://glossterm.org/index.php>
4. Белаичук А. Кратчайший путь к DocBook [Электронный ресурс]. Режим доступа к ресурсу: <http://docbook.ru/doc/sw/>
5. Технический писатель. Базовые компетенции специалиста [Электронный ресурс]. режим доступа к ресурсу: <http://philosoft.ru/twskills.zhtml>
6. Технология DITA. Обзор возможностей и основные преимущества [Электронный ресурс]. режим доступа к ресурсу: <http://www.philosoft.ru/dita-intro.zhtml>

7. Н.Уолш. DocBook: The Definitive Guide [Электронный ресурс]. Режим доступа к ресурсу: <http://www.docbook.org/tdg/en/html/docbook.html>