

# Автоматизация процесса формирования комплектов технической документации на основе онтологического подхода

## Process automation of formation technical documentation kit on base of the ontological approach

**Ключевые слова:** документирование – documentation; документация на программное обеспечение – software documentation; принцип единого источника – principle of single source; система управления знаниями – knowledge management systems; онтология – ontology.

Рассматривается принцип единого источника в документировании. Описывается фрагмент онтологии документации на программное обеспечение. Предложен подход формирования комплектов документации на программное обеспечение из фрагментов онтологии документации.

The article reviews the principle of single source in documentation. Example of ontology of software documentation fragment is described. The approach of formation software documentation kit from ontology fragments of documentation is offered.

### ВВЕДЕНИЕ

Документация на программное обеспечение – это печатные и электронные руководства, справочная система, справочный текст, сопровождающие некоторое программное обеспечение – программу или программный продукт. Эти документы описывают то, как работает программа и/или то, как ее использовать.

Документация является органической, составной частью программного продукта. Программы могут стать программным продуктом только в совокупности с комплектом документов, полностью соответствующих их содержанию и достаточных для его освоения, применения и изменения. Ошибки в документации не менее опасны, чем ошибки в структуре, интерфейсах и текстах программ. Поэтому к разработке, полноте, корректности и качеству документации необходимо такое же тщательное отношение, как к

**ДМИТРИЕВ / DMITRIEV P.**

**Павел Игоревич**

(Pavel.Dmitriev@billing.ru)  
соискатель кафедры информатики  
и информационных технологий СПбГУКИ,  
ведущий инженер ЗАО «Петер-Сервис»,  
Санкт-Петербург

разработке и изменениям текстов программ и данных.

В работе [4] рассматривается разработка системы управления знаниями (СУЗ) инфокоммуникационной системы, в которой документация на программное обеспечение (ПО) является основным ресурсом знаний такой системы. Основная задача СУЗ – накапливать не разрозненную информацию, а структурированные, формализованные знания. Основная цель СУЗ – сделать знания доступными и повторно используемыми.

При разработке системы управления знаниями (СУЗ) выделяют следующие этапы [2, 8]:

- накопление – стихийное и бессистемное накопление информации в организации;
- структурирование – на этом этапе должны быть выделены основные понятия, выработана структура представления информации, обладающая максимальной наглядностью, простотой изменения и дополнения;
- формализацию – представление структурированной информации в форматах машинной обработки, т.е. на языках описания данных и знаний;
- извлечение – процесс, идентичный традиционному извлечению знаний для экспертной системы (один из наиболее сложных и трудоемких этапов, от его успешности зависит дальнейшая жизнеспособность системы);
- обслуживание – под процессом обслуживания понимается корректировка формализованных данных и знаний (добавление, обновление), удаление устаревшей информации.

Для системы управления знаниями инфокоммуникационной системы типовые этапы разра-

ботки претерпевают некоторые изменения. Этап накопления отсутствует в такой СУЗ. Этапы структурирования и формализации сводятся к разработке онтологии документации на программное обеспечение и ее наполнению. Этап обслуживания остается без изменений. Особенности процесса извлечения знаний подробно описаны в работах [3, 4]. Дополнительным этапом СУЗ инфокоммуникационной системы является формирование комплектов документации. Рассмотрим данный этап более подробно.

### ПРИНЦИП ЕДИНОГО ИСТОЧНИКА

В технической документации повтор — явление не менее редкое, чем в тексте программы. Приходится описывать сходные функции программ и систем, сходные формы и сходные действия пользователей. Описания одних и тех же объектов приходится полностью или частично дублировать в разных документах. Поскольку материал, из которого состоит техническая документация, — текст на естественном языке, применить к нему модульный подход оказывается сложнее, чем к программному коду. В нем хватает нерегулярностей, обусловленных лексическими и грамматическими исключениями, традициями, соображениями стилистики и эстетики [6].

Принцип единого источника и реализующие его технологии позволяют применить модульный принцип к документированию. Применение технологии единого источника требует подготовки. Сначала проектируется структура единого источника, разрабатываются шаблоны и стили оформления, устанавливается и настраивается инструментарий для формирования документов. Таким образом, наблюдается связь данного этапа с этапом структурирования и формализации в СУЗ инфокоммуникационной системы.

Принцип единого источника в документировании, как и модульный принцип в программировании, помогает организовать работу коллектива документаторов, распределив между ними изолированные подзадачи. Таким образом, единый источник — это не только техническое, но еще и организационное решение. Формирование документов на основе единого источника — основной принцип, на котором сегодня построены все наиболее известные и развитые инструментальные средства для автоматизации документирования: AuthorIT, DITA, DocBook/XML, Rational SoDA и т.д. [1, 5, 7, 9]. Второе важное свойство технологий единого источника заключается в возможности придать одному и тому же тексту разное оформление и конвертировать его в любой нужный электронный формат.

Применение технологии единого источника в особенности целесообразно при разработке и длительном сопровождении комплектов документации на сложные решения — программные и программно-аппаратные комплексы, автоматизированные системы, большие корпоративные ИТ-инфраструктуры. Документирование таких решений в особенности требует эффективности при создании и обновлении наборов типизированных документов, описывающих типизированные предметы.

При документировании крупных комплексов и систем исключительно важной становится масштабируемость комплекта документации, возможность быстро распространить его на новые предметы (компоненты, подсистемы, пользовательские роли, операции, процессы) и быстро освоить выпуск новых типов документов. В настоящее время автоматизированные системы разработки технической документации, построенные на основе принципа единого источника, внедряются и успешно применяются в компаниях самого разного масштаба и профиля.

### ФОРМИРОВАНИЕ КОМПЛЕКТОВ ДОКУМЕНТАЦИИ НА ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Этап формирования комплектов документации инфокоммуникационной системы может быть реализован в двух вариантах. В основе первого варианта лежит интеграция СУЗ инфокоммуникационной системы и готового средства для автоматизации документирования. Второй вариант реализуется в рамках СУЗ. Второй вариант является более предпочтительным, так как минимизируются затраты на покупку готового решения, эксплуатацию и сопровождение.

В разрабатываемой СУЗ вся документация разделяется на фрагменты, которые представляют собой элементы знаний. Например, фрагменты — назначения подсистем, назначения продуктов, описания настроенных параметров, описания операций, описания программных методов и интерфейсов и т.д. Такой подход предполагает, что каждый документатор не разрабатывает цельные руководства, а создает фрагменты документации, помещает их в единый источник (библиотеку документов), указывает семантические свойства и связи с другими объектами.

На рисунке 1 представлен фрагмент онтологии документации на ПО для настроечного параметра. Основными понятиями являются «настроечный параметр», «функциональность» и «подсистема», именно они и будут классами в нашем фрагменте онтологии. Класс «Настроечный параметр» имеет следующие атрибуты (в

# БЕЗОПАСНОСТЬ

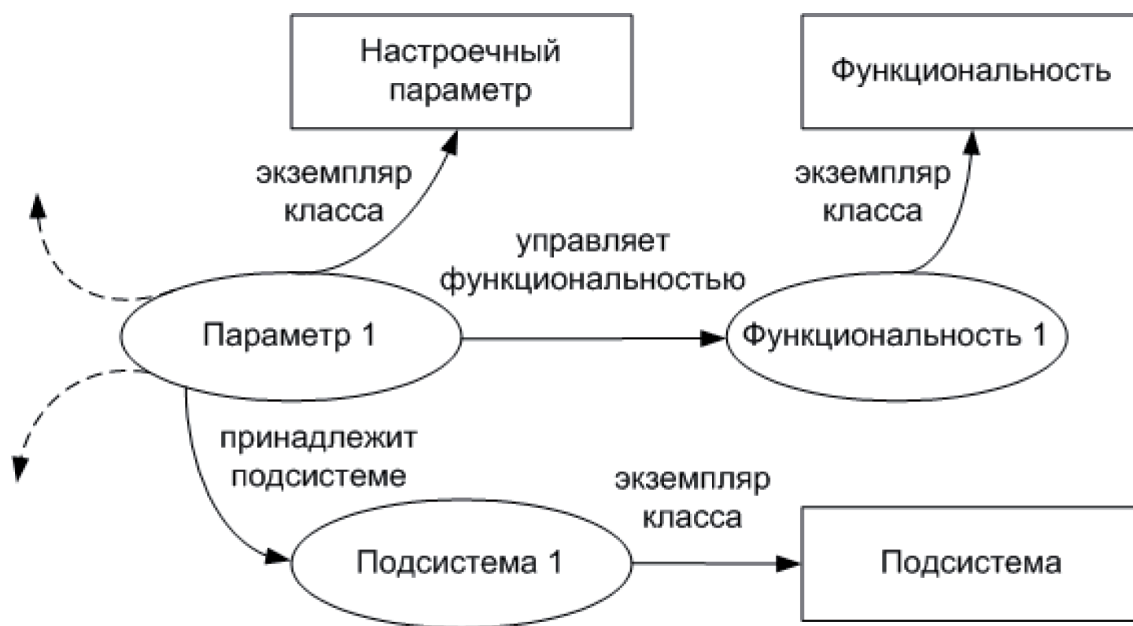


Рис. 1. Фрагмент онтологии документации на ПО для настроечного параметра

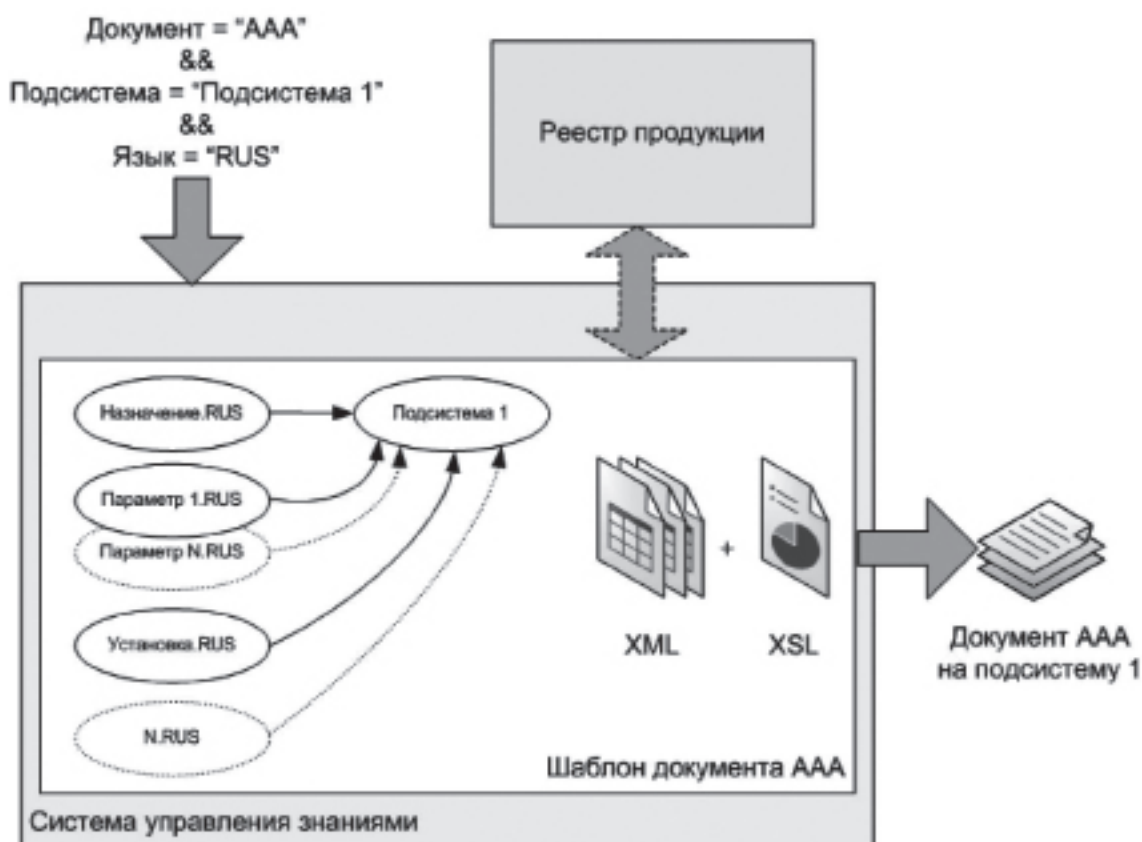


Рис. 2. Схема компоновки документа

скобках указаны тип значения и разрешенные значения атрибута):

- название (строка);
- подсистема (экземпляр класса «Подсистема»);
- функциональность (экземпляр класса «Функциональность»);
- язык (строка; возможные значения – русский, английский).

Заказчику отгружаются цельные документы, собранные на основе шаблонов из фрагментов. Формирование комплектов документации – процесс автоматизированной компоновки документов на ПО из фрагментов. Часть документов строится на основе predefined шаблонов документов. Другая часть документов собирается на основе выборки, полученной пользователем на этапе извлечения знаний либо непосредственно им указанной. На рисунке 2 представлена схема компоновки документа из фрагментов онтологии документации на ПО.

Рассмотрим пример формирования руководства системного программиста на подсистему и справочника настроенных параметров на продукт на основе predefined шаблонов документов. Входными данными для шаблона руководства системного программиста на подсистему являются наименование подсистемы и язык документа. Данные атрибуты задаются для каждого класса онтологии документации на ПО. Процесс формирования руководства системного программиста на подсистему состоит из следующих шагов:

- получение фрагментов по подсистеме и языку;
- вывод на экран фрагментов в определенной последовательности и стилевом оформлении;
- сохранение документа в требуемом формате.

Этапы вывода и сохранения могут быть объединены.

Входными данными для шаблона справочника настроенных параметров на продукт являются наименование продукта и язык документа. Процесс формирования справочника настроенных параметров на продукт состоит из следующих шагов:

- определение совокупности наименований входящих в продукт подсистем (запрос информации в реестре продукции);
- получение фрагментов – настроечных параметров по подсистемам и языку;
- вывод на экран;
- сохранение документа.

## ЗАКЛЮЧЕНИЕ

Онтология документации играет решающую роль в модели описания знания инфокоммуникационной системы. Имеющиеся знания в онтологии документации позволяют автоматизировать выпуск комплектов документов без использования дополнительных программных средств. Использование принципа единого источника при разработке документации на ПО позволяет снизить трудоемкость разработки и сопровождение документации, а также повысить качество самой документации.

## Литература

1. Автоматизация разработки технической документации с применением AuthorIT. – [www.authorit.ru/pdf/book.pdf](http://www.authorit.ru/pdf/book.pdf).
2. Гладун А.Я. Онтологии в корпоративных системах // Корпоративные системы. – 2006. – № 1. – С. 41–47.
3. Дмитриев П.И. Основные ресурсы системы управления знаниями инфокоммуникационной системы // Мат-лы Международной научно-технической конференции, посвященной 80-летию СЗТУ «Системы и процессы управления и обработки информации». – Ч. 1. – СПб.: СЗТУ, 2010. – С. 182–189.
4. Дмитриев П.И. Разработка системы управления знаниями инфокоммуникационной системы // Мат-лы XI Международной научно-практической конференции молодых ученых, студентов и аспирантов «Анализ и прогнозирование систем управления». – Ч. 1. – СПб.: СЗТУ, 2010. – С. 135–140.
5. Новичков А. Система генерации проектной документации Rational SoDA. – [www.compress.ru/Article.aspx?id=12028](http://www.compress.ru/Article.aspx?id=12028).
6. Острогорский М. Разработка технической документации на основе единого источника. – [www.philosoft.ru/sssth-01.shtml](http://www.philosoft.ru/sssth-01.shtml).
7. Острогорский М. DocBook/XML: открытая платформа для разработки технической документации. – [www.philosoft.ru/docbook-basics.shtml](http://www.philosoft.ru/docbook-basics.shtml).
8. Попов Э.В. Управление знаниями. – [www.rfbr.ru/pics/20742ref/uprznan.pdf](http://www.rfbr.ru/pics/20742ref/uprznan.pdf).
9. Day D. Introduction to the Darwin Information Typing Architecture. – [www.ibm.com/developerworks/xml/library/x-dita1/](http://www.ibm.com/developerworks/xml/library/x-dita1/).