



УДК 004.82

Шуйский Илья Юрьевич

РАЗРАБОТКА СИСТЕМЫ УПРАВЛЕНИЯ ЗНАНИЯМИ ДЛЯ РАБОТЫ С ДОКУМЕНТАЦИЕЙ НА ОСНОВЕ KNOWLEDGE.NET

Аннотация

В статье освещаются существующие принципы организации документации на разрабатываемое программное обеспечение, особое внимание уделяется системам управления знаниями и онтологическому подходу. Приводятся существующие системы для работы с документацией, отмечаются их преимущества и недостатки и подчеркивается нерешенность проблемы организации документации.

Описывается концепция и методы реализации системы управления знаниями для работы с документацией на основе технологии Knowledge.NET, разработанной на кафедре информатики СПбГУ под руководством проф. В.О. Сафонова.

Ключевые слова: управление знаниями, документация, программное обеспечение, Knowledge.NET.

1. ВВЕДЕНИЕ

В современном мире разработка программного обеспечения (ПО) является ведущей отраслью. Информационные технологии используются повсеместно, сейчас трудно представить сферу человеческой деятельности, полностью независимую от них. В любых компаниях, каким бы родом деятельности они ни занимались, от производства корма для животных до производства сложных расчетов в банковской сфере, используются информационные системы.

Для того чтобы информационная система могла использоваться с успехом, должен пройти длительный процесс. В начале некая ИТ-компания разрабатывает систему, после чего внедряет ее. На этом деятельность компании-разработчика не заканчива-

ется – ведь при эксплуатации возникает множество проблем: в системе возникают ошибки, необходимо совершать доработки функциональности, обновлять программное обеспечение. Все это решается в рамках технической поддержки, которую компания-разработчик при необходимости оказывает заказчику. Однако есть еще одна проблема, которая возникает как перед заказчиком, который эксплуатирует систему, так и перед разработчиком, который ее изменяет, настраивает и поддерживает: необходимо знать, как система работает. Данная статья посвящена вопросу решения именно этой проблемы.

В разработке сложных систем участвует большое количество программистов, которые нередко находятся в разных городах и даже странах. Коллектив разработчиков со временем меняется – одни сотрудники уходят, другие, наоборот, появляются. Для того чтобы процесс разработки был эффективен

© Шуйский И.Ю., 2013

и не зависел от конкретных сотрудников, необходима подробная документация на ПО. Такая документация нужна заказчику не меньше, ведь пользователей системы также может быть много, они могут находиться в разных местах и меняться со временем.

Эффективность и «полезность» документации всецело зависит от удобства ее использования. Существует несколько подходов к ее созданию:

Текстовые документы

Самый простой и часто используемый способ построения системы документирования ПО – создание текстовых документов (формат doc или docx). Это может быть как один большой документ, так и набор документов меньшего размера. При работе с крупным документом (например, в тысячу страниц) возникают проблемы поиска нужной информации, а также соотнесения и анализа связанной информации. При работе с набором документов это также осложняется проблемой выбора нужного документа.

Документы с гиперссылками

Такой подход часто используется во встроенной документации, которая часто называется help (помощь). Это встроенная в систему подсказка, которая представляет собой набор текстовых документов, связанных друг с другом гиперссылками. Это гораздо удобнее, чем простая текстовая документация: ведь для того чтобы посмотреть связанную информацию или просто узнать о чем-то более подробно, достаточно перейти по гиперссылке, и сразу же открывается нужный раздел. Однако у этого подхода также есть недостатки. Во-первых, таким образом легко можно организовать пользовательскую документацию, но гораздо сложнее организовать документацию для администраторов и разработчиков. Во-вторых, проблема поиска нужной информации остается – это, по-прежнему, либо простой текстовый поиск, либо бесконечное хождение по гиперссылкам без какой-либо гарантии получить полную и исчерпывающую картину по интересующему вопросу.

Система управления знаниями (СУЗ)

К этому подходу относятся системы, организованные интеллектуальным образом. Содержащиеся в них данные являются уже не просто информацией, – они превращаются в знания. Вопрос различия понятий «знание» и «информация» подробно рассмотрен в статьях [1] и [3]. Существует несколько основных методов представления и описания знаний. Рассмотрим основные из них.

Продукционная модель. Эта модель основана на правилах вида «если (условие) – то (действие)», под условием понимается некоторый образец, по которому производится поиск в имеющейся базе знаний, в случае успешного поиска выполняется действие из правила. Такая модель часто используется в промышленных экспертных системах для определения действий персонала в определенных условиях.

Семантическая сеть. Представляет собой ориентированный граф, вершинами которого являются понятия (объекты), а дуги – отношения между ними. Характерными являются отношения типа класс-экземпляр, свойство-значение свойства и фрагмент-элемент класса. В качестве примеров отношений можно привести отношения часть-целое, функциональные отношения («производит», «определяет»), количественные (больше, меньше, равно), пространственные, временные, атрибутивные свойства и т. п. Семантические сети используются при анализе и обработке текста на естественном языке, системах искусственного видения и других. Также семантическая сеть является основой для других моделей представления знаний.

Фреймы. Фрейм является абстрактным образом для представления некоего стереотипа восприятия. По своей структуре фрейм состоит из имени и некоего набора слотов, заполненных определенными значениями.

Онтология. Слово онтология пришло из древнегреческого языка (онтос – сущее, логос – учение, понятие). Это термин, определяющий учение о сущем, бытии, в отличие от гносеологии – учении о познании. Существует несколько определений понятия он-

тология применительно к управлению знаниями, например: «Онтология – это формальное, явное, точное определение (спецификация) совместно используемой концептуализации [Gruber T.A., 1995], «Онтология – это базы знаний специального типа, которые могут «читаться» и пониматься, отчуждаться от их разработчика и/или физически разделяться их пользователями» [Гаврилова Т.А., Хорошевский Ф.В., 2001]. С практической точки зрения, онтология состоит из терминов, организованных в таксономию, их определений, атрибутов, а также связанных с ними аксиом и правил вывода. В каком-то смысле, концепция онтологии включает в себя и концепцию фреймов как неких объектов с атрибутами и концепцию семантической сети как представление объектов и отношений между ними.

Онтологии, таким образом, позволяют представить знания о какой-то области в виде, понятном человеку, с одной стороны, и достаточно формализованном, чтобы они могли обрабатываться компьютером, – с другой. Благодаря этому, в последнее время именно онтологический подход считается наиболее перспективным и часто используемым в разработке систем управления знаниями.

Среди перечисленных подходов к построению документации самыми эффективными, удобными и интересными являются системы управления знаниями. В современном мире, когда корпоративные информационные системы становятся все больше и сложнее, а количество их пользователей постоянно увеличивается, проблема эффективного получения знаний о существующих системах встает особенно остро. Обычные хранилища текстовой информации здесь не подходят, так как они не решают проблемы поиска нужных знаний. Необходимы механизмы организации информации, которые бы позволяли хранить знания и эффективно находить ответы на интересующие вопросы. Знания в таких системах должны храниться в виде, понятном человеку, так как он является конечным пользователем системы, и, в то же время, быть достаточно сильно формализованы, чтобы быть обработанными

компьютером. Онтологический подход к построению системы управления знаниями наиболее полно удовлетворяет этим требованиям.

2. СУЩЕСТВУЮЩИЕ СИСТЕМЫ ОРГАНИЗАЦИИ ДОКУМЕНТАЦИИ

На данный момент существует большое количество систем, предназначенных для работы с документами в том или ином виде. Приведем некоторые из них, указав их особенности и сферы использования.

Siemens Teamcenter (ST). Система управления документацией на протяжении жизненного цикла продукта. Позволяет создавать документацию из компонент различного формата (изображения, XML, текстовая информация и др.) и публиковать ее. ST предоставляет возможности поиска информации, записанной в различных форматах, и версионного контроля.

http://www.plm.automation.siemens.com/en_us/products/teamcenter/index.shtml

Manage Point (MP). Система управления документами внутреннего использования, предоставляет единую точку для работы с информационными каналами на предприятии, такими как: электронная почта, факсы, текстовые документы и др. MP позволяет осуществлять такие действия с документами, как поиск по тексту и различным атрибутам, загрузка и выгрузка в различных форматах.

<http://www.managepoint.com.au/index.html>

Filehold (FH). Система для работы с документами на протяжении их жизненного цикла. FH позволяет создавать документы, перенаправлять их на определенных сотрудников в рамках определенных workflow, находить документы по тексту и метаданным и др. Является системой внутреннего использования для управления библиотекой загруженных документов.

<http://www.filehold.com/help/home>

Logical Doc, BMS – системы внутреннего использования для работы с документами на предприятии, аналогичные Filehold.

Knova. CRM-система для работы с обращениями, поступающими по различным ка-

налам. В качестве единицы представления информации используются «разрешения» (resolutions), в системе также предусмотрены возможности создания различных рабочих процессов (workflow) для работы с обращениями. Knova обрабатывает поисковые запросы на естественном языке, что делает ее интерфейс удобным для клиентов.

<http://www.consona.com/knova/>

Smart Support. Система оперирует понятием «статья» (article) – предоставляет возможности создания, категоризации и поиска статей. Имеется возможность гибкой настройки интерфейса, а также создания уведомлений, управление правами пользователей и подключение дополнительной функциональности – форум, чат и др. Smart Support имеет очень простую структуру и использует CRM-систему для поиска информации. Может быть развернута на платформах Windows, Mac, Linux/Unix, а также на веб-сайте.

<http://www.safeharbor.com/solutions/smartsupport/>

Bitrix Intranet. Система организации рабочего процесса внутри компании – позволяет организовать обмен файлами между пользователями. Bitrix Intranet предоставляет множество каналов коммуникации между пользователями: почта, видео-звонки, чат. Для всех пользователей создаются и настраиваются профили, которые могут включаться в различные рабочие группы. Система Bitrix Intranet используется как корпоративная система для обмена информацией и организации рабочего процесса внутри компании. Bitrix Intranet поддерживает платформы Windows, Mac, Linux/Unix.

<http://www.bitrixsoft.com/products/intranet/>

Globalbrain. Система обеспечивает управление различными типами контента внутри организации, позволяет структурировать работу с электронными документами, почтой, мультимедиа-контентом, XML-файлами и информацией другого формата. GlobalBrain используется для работы с дебиторской/кредиторской задолженностью, для управления заказами и в других CRM-системах. Система может работать на архитектуре Windows или на базе веб-сайта.

<http://www.perceptivesoftware.com>

Interspire Knowledge Manager (IKM). База знаний корпоративного класса, позволяет хранить и обмениваться информацией как внутри компании, так и с заказчиками и партнерами. Информация хранится в виде документов Microsoft Word и PDF. Есть возможность категоризации и поиска информации, также система предоставляет широкие возможности настройки доступа и кастомизации интерфейса. Доступ к ИКМ, как правило, организуется через вебсайт.

<http://www.interspire.com/knowledgemanager/>

Oracle Knowledge Management System (OKMS). Система управления знаниями компании oracle позволяет хранить информацию в виде «решений» (solutions), состоящих из наборов текстовых утверждений с прикрепленной информацией другого формата. Имеются различные возможности поиска решений и утверждений – по словам, категориями типам. Также система поиска выдает связанные решения и определяет их определенную «релевантность». OKMS интегрируется с другими продуктами компании oracle – TeleService, DepotRepair, iSupport и др. и используется как хранилище данных для различных CRM-систем.

<http://www.oracle.com/us/products/applications/oracle-knowledge-management-1352098.html?ssSourceSiteId=opn>

Knowledge Spaces (KS). CRM-система на базе интернет-портала. Информация в системе хранится в виде статей, к которым могут быть прикреплены документы других форматов. Поиск в KS осуществляется с учетом категорий и релевантности результатов.

<http://www.moxiesoft.com/products/knowledgebase.aspx>

Solution Manager (SM). Система управления знаниями, позволяющая создавать не только статьи, но и интерактивные решения, например деревья принятия решений, в которых конечный пользователь, шаг за шагом отвечая на определенные вопросы, приближается к решению своей проблемы. SM может использоваться как внутренняя СУЗ, но, в первую очередь, как внешняя CRM-система. Поддерживает платформы Windows, Linux/Unix и размещение на веб-портале.

<http://rightanswers.com/products-knowledge-management-solutions-manager.htm>

3. СУЩЕСТВУЮЩАЯ ПРОБЛЕМА

Перечисленные выше системы можно разделить на две основных категории:

1. Системы организации внутреннего документооборота компании, которые позволяют обрабатывать информацию различного рода (почта, документы, медиа) в единой точке.

2. CRM-системы, предоставляющие некие возможности поиска текстовой информации (ключевые слова, категоризация).

Такие системы крайне неэффективны для организации документации на сложные программные комплексы, где необходимо учитывать большое количество настроек, параметров установки, взаимных зависимостей компонентов и их функциональности.

Также существует большое количество исследовательских проектов разработки систем управления знаниями, основанных на онтологиях. Один из самых известных из них – проект Protégé, универсальная система для построения онтологий и баз знаний. Однако все эти системы не предназначены для промышленной реализации СУЗ, а носят скорее научно-исследовательский характер.

Таким образом, можно сделать вывод, что на данный момент проблема работы с документацией на разрабатываемое программное обеспечение не решена в полной мере. То есть, не существует системы, которая позволяла бы организовать сложную документацию в виде, удобном для поиска нужной информации, в то время как проблема такой организации встает все более и более остро.

4. ПРЕДЛАГАЕМОЕ РЕШЕНИЕ

Для решения указанной проблемы предлагается создать систему управления знаниями, в основе которой лежит онтологический подход. Опишем подробнее внутреннюю структуру хранения информации этой системы.

В качестве внутреннего наполнения системы выступает документация, то есть информация, представленная в виде текста с

изображениями. Весь материал разделен на блоки, которые в данной системе считаются атомарными. То есть, забегая вперед, получить нужную информацию система позволяет с точностью до блока, в которой эта информация находится. Такой блок должен содержать исключительно конкретную и неделимую информацию, как, например, функция настроечного параметра, описание элемента визуального интерфейса или назначение некоей подсистемы и т. д. Грамотное разбиение самой документации на такие блоки может значительно повысить (а неграмотное – понизить) эффективность системы, изучение этого вопроса может стать предметом отдельных исследований, выходящих за рамки темы данной работы, поэтому мы не будем уделять ему пристальное внимание. Важно сказать, что такие блоки должны быть достаточно мелкими, чтобы каждый из них содержал одну «единицу» информации, и в то же время достаточно крупными, чтобы эта «единица» содержалась в нем целиком. В таком случае поиск нужной информации сводится к поиску одного такого блока или некоторого набора блоков.

Каждый блок, помимо своего основного содержания, имеет определенную метаинформацию, иначе говоря – свойства, в частности:

- Тип блока, что в нем содержится – то самое описание настройки или элемента визуального интерфейса.
- Функциональная зависимость от других блоков, например, какие настройки влияют на данный блок. Зависимости также могут быть различных типов.
- Управляемая функциональность – информация, обратная к предыдущему пункту.
- Является частью – то есть рассматриваемый блок является частью некоторой более крупной сущности
- Включает – обратное свойство к предыдущему.

....

Выше был приведен неполный список возможной метаинформации, относящейся к блоку документации. Набор таких свойств зависит от конкретной разрабатываемой продукции и от содержания самой документа-

ции. В процессе создания документации необходимо определить перечень таких свойств и наделять ими каждый блок, загружаемый в систему. Пример структуры данных приведен на рис. 1.

Заметим, что отношение включения порождает таксономию на получившейся онтологии, в то время как отношение зависимости является транзитивным, это также следует учитывать при поиске и отображении нужной информации.

Вместе с такой метаинформацией, содержимое блоков документации из простой текстовой информации превращается в нечто большее – в знания. Метаинформация позволяет выявить то, что не описано в документации в явном виде, то есть показать так называемые неявные знания. Именно наличие таких знаний отличает СУЗ от обычного массива текстовой информации и делает ее более ценной и эффективной.

Описанные блоки документации, вместе с метаинформацией и отношениями образуют онтологию документации на разрабатываемое программное обеспечение. Эта онтология хранится в разрабатываемой системе управления знаниями в базе документации, интерфейсом для которой служит модуль онтологии (МО). Вся работа с документацией осуществляется через МО, этот модуль обеспечивает необходимый уровень абстракции, то есть превращает набор бло-

ков документации в онтологию, с которой удобно работать.

Однако сама по себе информация не представляет никакой ценности, если нет механизмов для работы с ней. Основные действия, которые необходимо производить с информацией, – это добавление новой информации и поиск нужной информации в существующей. Для реализации этой функциональности в системе существует еще 2 модуля: модуль загрузки и модуль поиска. Рассмотрим их подробнее.

Модуль загрузки. Модуль загрузки (МЗ) предназначен для размещения в СУЗ новых блоков документации. Он позволяет загрузить в систему блок и определить его метаинформацию, то есть если рассматривать свойства, приведенные выше, то указать тип добавляемого блока, а также множество существующих блоков, связанных с ним отношениями зависимости и включения. Функциональность модуля загрузки фактически сводится к добавлению нового блока и заполнению его свойств с метаинформацией.

Модуль поиска. Модуль поиска (МП) является ключевым модулем системы. Он позволяет найти нужную информацию из огромного набора загруженных модулей. МП осуществляет не только текстовый поиск информации, но также отслеживает отношения между модулями (включение и зависимость), что позволяет не просто привести

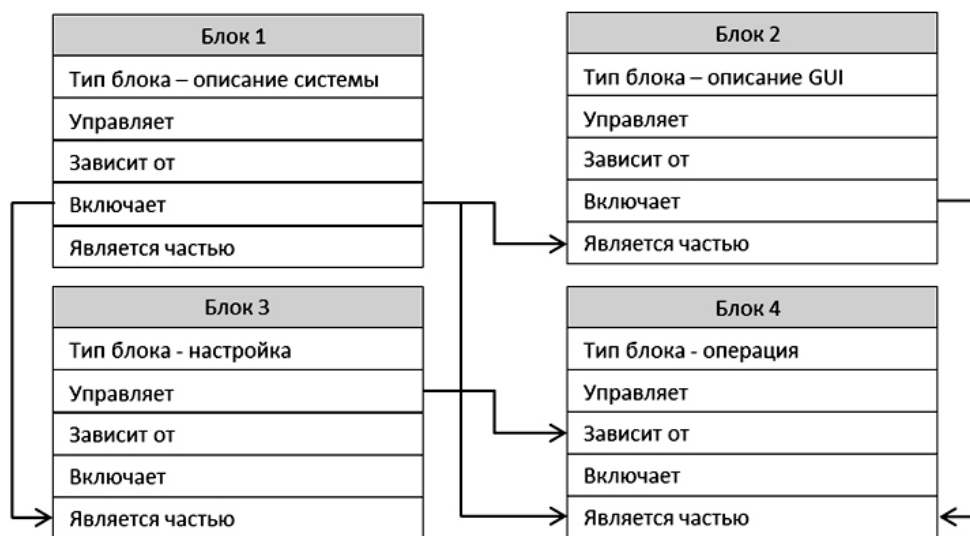


Рис. 1. Структура онтологии

сухой кусок текста, в котором упоминаются искомые слова, но получить целостную картину по интересующему вопросу. Впоследствии возможна фильтрация полученного набора блоков по значениям различных свойств, эта функциональность относится также к МП.

Таким образом, СУЗ состоит из трех основных компонентов: модуля документации, модуля загрузки и модуля поиска. Структура взаимодействия модулей представлена на рис. 2.

Таким образом, получается система, состоящая из трех модулей и базы данных. Опишем функции каждого модуля, необходимые для взаимодействия внутри системы и успешного функционирования СУЗ в целом.

Модуль загрузки:

- Загрузка одного блока документации (в общем случае – с текстовой информацией) в базу через МО. При этом прописываются свойства блока, его метаданные (в частности, ссылки на связанные блоки).

Модуль онтологии:

- Помещение блока в базу документации.
- Извлечение блока из базы документации.

Модуль поиска

- Поиск модуля по набору слов.
- Поиск всех модулей, связанных с найденным.
- Расположение модулей по релевантности.

Следует отметить, что такое разделение СУЗ на модули пока носит скорее условный характер – это различные группы функций, которые физически могут быть реализованы и в одной компоненте, и в нескольких.

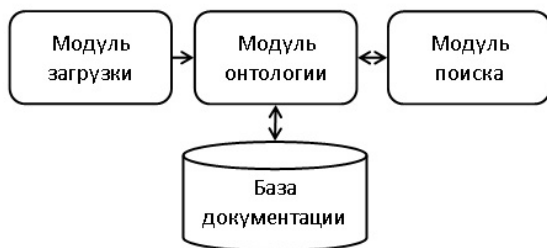


Рис. 2. Компоненты СУЗ

Поэтому мы будем называть их «функциональными модулями».

5. РЕАЛИЗАЦИЯ

Для реализации описанной системы необходим механизм, который бы, с одной стороны, имел все преимущества современных средств программной инженерии, а с другой стороны, предоставлял возможности инженерии знаний. В качестве такого механизма была выбрана технология Knowledge.Net, разработанная на математико-механическом факультете СПбГУ под руководством проф. В.О. Сафонова. Технология Knowledge.Net включает в себя одноименный язык представления знаний, который является расширением языка C# и легко интегрируется с платформой .NET. Язык оперирует понятиями *концепт*, *экземпляр*, *свойство* и др., на которые крайне удачно ложится терминология описанной задачи, ведь язык изначально был разработан для описания онтологий и работы с ними. Кроме того, в нем имеется ряд преимуществ, связанных с транзитивностью свойств и логическим выводом, что очень полезно в решаемой задаче.

Опишем подробнее реализацию СУЗ с помощью конструкции языка Knowledge.Net.

Блок документации реализуется с помощью концепта *Doc_unit*. Так как в Knowledge.NET все определяемые концепты являются конкретизацией концепта *Thing*, это будет записано следующим образом:

Doc_unit is_subconcept_of Thing;

Метаинформация, или свойства блоков, задаются с помощью свойств концептов. В Knowledge.NET поддерживается несколько видов свойств, среди них есть простые свойства (связывают экземпляр со значением какого-то простого типа данных) и объектные свойства (связывают между собой два экземпляра).

Каждое свойство в Knowledge.NET имеет определенный домен и область значений, которые задаются с помощью ключевых слов *domain* и *range* соответственно. Про-

стые свойства блоков документации задаются следующим образом (на примере свойства «номер версии»):

```
datatype property HasVersion
{
  Domain doc_unit;
  Range int;
}
```

Для реализации более сложных свойств объектов, таких как функциональная зависимость, используются объектные свойства. Для указания их транзитивности используется ключевое слово *transitive*. Также для реализации обратных свойств используется соответствующая конструкция языка Knowledge.NET, обратные свойства при этом также будут транзитивными. Описания свойства «управляет функциональностью» вместе с обратным к нему свойством «зависит от»:

```
transitive object property Controls
{
  Domain doc_unit;
  Range doc_unit;
  Inverse Depends;
}
```

Свойство «включает» вместе с обратным к нему «является частью» описывается аналогично:

```
transitive object property Includes
{
  Domain doc_unit;
  Range doc_unit;
  Inverse IsPartOf;
}
```

Для реализации различных типов блоков, таких как описание настройки, алгоритма работы, операции, порядка установки и т. д. используется также конструкция под-концепта:

```
Setting is_subconcept_of Doc_unit;
Description is_subconcept_of Doc_unit;
Operation is_subconcept_of Doc_unit;
Installation is_subconcept_of Doc_unit;
```

Поиск документов реализуется с использованием возможностей встроенного языка запросов в Knowledge.NET. Язык запросов позволяет выбирать экземпляры из онтоло-

гии, которые удовлетворяют определенным критериям, и имеет следующий синтаксис:

```
Individuals of concepts CONCEPT1,
CONCEPT2...
where Expr I;
```

Запросы в Knowledge.NET дают широкие возможности для логического вывода, однако возникают сложности при реализации поиска с использованием транзитивных свойств (например, вывести зависимые блоки документации). Это одна из проблем, которые возникают при разработке описываемой СУЗ, но не являются неразрешимыми.

Ценность знаний зависит не только от их содержания и удобства поиска. Важное значение имеет их доступность. Широко распространенный в настоящее время интернет делает доступной хранящуюся в нем информацию, поэтому наиболее эффективной моделью реализации СУЗ будет веб-сервер, то есть узел, доступ к которому может быть осуществлен через интернет. Действительно, СУЗ, находящаяся локально на рабочем компьютере одного сотрудника, фактически не представляет никакой пользы. Гораздо эффективнее система, которая развернута на внутреннем сервере компании, тогда все ее сотрудники получают доступ к знаниям. Однако в таком случае СУЗ не может быть использована персоналом заказчика, а если речь идет о документации на сложные программные комплексы, настройку и администрирование которых делает заказчик (хотя бы частично), то этот аспект становится чрезвычайно важным.

СУЗ, развернутая на веб-сервере, будет являться частью крупного интернет-портала, структура и дизайн которого определяются компанией-владельцем. Таким образом, визуальный интерфейс СУЗ не представляет ценности. Наоборот, наличие хорошего API будет очень удобным решением, для разворачивания на сервере и интеграции в его интерфейс. API также можно использовать для модификации системы «под себя», результат вызовов API-функций может быть дальше обработан другими процедурами.

6. ЗАКЛЮЧЕНИЕ

В статье описана концепция и принципы реализации системы управления знаниями для работы с документацией на основе Knowledge.NET. Проблема организации работы с документацией стоит достаточно остро, существующие подходы не могут в полной мере решить имеющиеся проблемы хранения и поиска информации. Описанная система позволит решить эти проблемы с помощью онтологической модели представления знаний и интеллектуального механизма поиска. В качестве реализации системы предлагается создание API, которое может

быть развернуто на веб-сервере и интегрировано в интерфейс интернет-портала, что обеспечит легкий доступ к содержащимся в системе знаниям.

В качестве технологии для реализации системы выбран язык Knowledge.NET – перспективный язык онтологического представления знаний, который является расширением языка C#. Выбор обусловлен тем, что технология Knowledge.NET, разработанная на математико-механическом факультете СПбГУ под руководством проф. В.О. Сафонова, позволяет использовать все преимущества программной инженерии вместе с возможностями инженерии знаний.

Литература

1. Сафонов В.О и др. Интеграция методов инженерии знаний и инженерии программ: система управления знаниями Knowledge.NET, 2005.
2. Сафонов В.О. и др. Knowledge.NET ontology-based knowledge management toolkit for Microsoft.NET, .NET Technologies. Плизань, 2006.
3. Тузовский А.Ф., Чириков С.В., Ямпольский В.З. Системы управления знаниями (методы и технологии). Издательство научно-технической литературы. Томск, 2006.
4. Кознов Д.В., Бугайченко Д.Ю. Введение в программную инженерию. Интернет-университет информационных технологий / intuit.ru, 2009 (дата обращения 20.02.2013).
5. Кознов Д.В., Смирнов М.Н., Дорохов В.А., Романовский К.Ю. WebMLDoc: подход к автоматизированному отслеживанию изменений в пользовательской документации Web-приложений. Вестник Санкт-Петербургского Университета, Сер. 10. Вып. 3. 2011. С. 112–126.
6. Гаврилова Т.А. Онтологический инжиниринг // Труды 8 национальной конф. по искусственному интеллекту. М.: Физматлит, 2002. С. 846–853.
7. Вебер А.В., Данилов А.Д., Шифрин С.И. Knowledge-технологии в консалтинге и управлении предприятием. М.: Наука и техника, 2002.
8. Web-страницы проекта Knowledge.net <http://www.knowledge-net.ru> (дата обращения 20.02.2013).
9. Web-портал <http://www.capterra.com/knowledge-management-software> (дата обращения 20.02.2013).
10. Web-страницы сайта <http://ru.wikipedia.org> (дата обращения 20.02.2013).

Abstract

The article covers the existing principles of organizing documentation to software under development. Special attention is paid to knowledge management systems and ontological approach. Existing systems for documentation development are analyzed, their advantages and shortcomings are shown, it is emphasized that the problem of organizing documentation is far from its solution.

The concept and the implementation methods are proposed for a documentation development toolkit based on the Knowledge.NET technology developed at the chair of computer science of SPbSU under the supervision of Professor Vladimir O. Safonov.

Keywords: knowledge management, documentation, software, Knowledge.NET.



Наши авторы, 2013.
Our authors, 2013.

*Шуйский Илья Юрьевич,
окончил аспирантуру СПбГУ,
математико-механический
факультет, кафедра информатики,
ilja.shujskij@gmail.com*