



Автоматизация разработки Интернет-информационных систем на базе OWL-онтологий

Введение

- Некоторые проблемы, возникающие в процессе разработки информационных систем
 - смена дизайна системы
 - Зачастую реализуемая система должна обладать некими стандартными стилем и дизайном, соблюдаемыми во всем приложении. Сменить дизайн просто только на начальных этапах разработки, в дальнейшем может вызывать ряд трудностей.
 - создание стандартного интерфейса пользователя для каждого типа ресурса
- Что нужно автоматизировать в этом процессе
 - создание и настройка визуального представления данных
 - обеспечение минимальной функциональности манипулирования данными
 - По сформулированной модели данных предметной области и соответствующим параметрам необходимо генерировать фрагменты системы/приложения реализующие основные операции работы с данными, а именно: создание и удаление объектов, просмотр и изменение свойств объекта, а также поиск по атрибутам объекта.

Требования к системе автоматизации разработки ИС

- Инструменты моделирования как предметной области, так и прикладной логики работы с данными
- Минимальное дублирование информации
 - Языки спецификации моделей должны предоставлять возможность повторного использования фрагментов моделей без их прямого копирования.
- Отсутствие платформенно-зависимых конструкций
 - Разрабатываемая технология не должна содержать понятий специфических для определенной среды выполнения приложений, поскольку по описанной модели данных могут быть сгенерированы различные виды приложений.

Предлагаемое решение

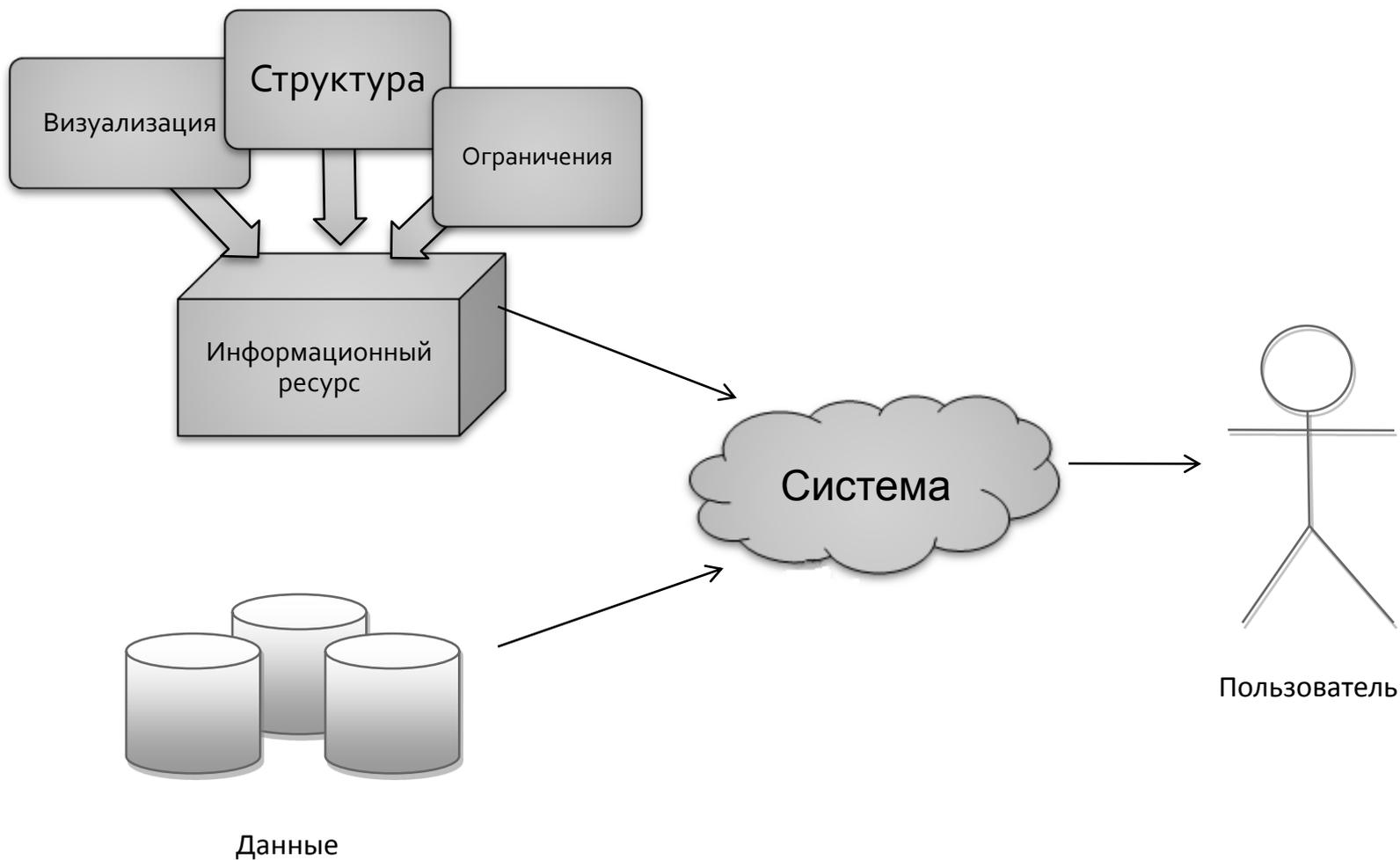
- **Идея**

- Данные характеризуются структурой и дополнительными метаданными (такими как параметры визуализации и ограничения)
- Ответ пользователю формируется с использованием сведений, заданных при разработке онтологии

- **Этапы моделирования**

- Описание структуры данных – описывается модель предметной области
- Создание модели взаимодействия с пользователем – описывается как именно должна быть представлена информация для конечного пользователя
- Определение ограничений целостности данных

Схема взаимодействия с пользователем



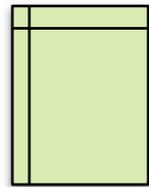
Реализация решения для платформы **Web**.

- Используемые технологии
 - Язык Java – создание прикладной логики приложения
 - Язык OWL – описание онтологии
- Постановка задачи в рамках **Web** платформы и определенных выше технологий:
 - Создание owl-ориентированной системы, обеспечивающей быструю разработку Интернет-информационных систем, ориентированных на хранение и представление информационных ресурсов

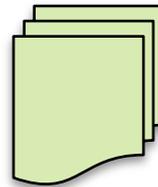
Реализация решения для платформы Web.

Визуализация

- **Идея:**
 - формализовать визуализацию объекта
- **Решение:**
 - разделить определение занимаемого местоположения свойств типа ресурса с их визуальным представлением
 - визуальное представление произвольного типа ресурса определяется визуальным представлением каждого его атрибута и их взаиморасположением ("**каркас**" визуализации)
 - визуальное представление информации в этом местоположении задается **шаблоном** визуализации



Каркас



Набор
шаблонов



Визуальное
представление

Реализация решения для платформы Web.

Визуализация

- Основные понятия:
 - **Каркас отображения** - определяет компоновку визуальной среды пользователя, может использоваться для отображения любого класса определенного в модели предметной области.
 - **Шаблон отображения** - определяет визуализацию конкретного атрибута или класса целиком (например, атрибут типа строка может быть визуализирован в виде поле ввода или же простым текстом).
 - **Тип отображения** - определяет соответствие между атрибутами классов и шаблонами отображения (например, при выводе атрибута типа строка на просмотр, будет использован шаблон, выводящий строку простым текстом, а при выводе на редактирование – в виде поля ввода)
 - По умолчанию доступны 3 типа отображения: «на просмотр», «на редактирование», «на поиск».

Пример настройки визуализации ресурса

- Каркасы визуализации

- *список полей объекта (property-sheet)* – представляет собой таблицу с 2-мя столбцами, первый столбец представляет название поля, второй значение. Обычно используется для вывода конкретного экземпляра класса.

Фамилия	Иванов
Имя	Иван
Отчество	Иванович

- *сетка свойств списка объектов (grid)* – таблица, столбцами которой ставятся в соответствие свойства класса. Каждый объект списка выводится в новой строчке таблицы. Обычно используется для вывода коллекции объектов.

Фамилия	Имя	Отчество
Иванов	Иван	Иванович
Петров	Петр	Петрович

Пример настройки визуализации ресурса

- Шаблоны визуализации
 - Шаблон определяет визуализацию конкретного класса или типа данных
 - Шаблон может обладать набором параметров, настраивающих визуализацию
 - Примеры шаблонов для типа данных строка (String). Ниже приводятся 2 шаблона отображения данных типа строка. Это отображение в виде элемента ввода (text) или в виде простого текста (label). Каждый шаблон имеет набор параметров, которые настраивают его отображение.

Название	Описание	Параметры	
text	HTML элемент input	class	CSS класс, который будет применен к элементу
		type	Тип поля, одно из списка: text, textarea, password, rich.
		style	CSS стиль, который будет применен к элементу
label	HTML элемент label, выводит текст	class	CSS класс, который будет применен к элементу
		style	CSS стиль, который будет применен к элементу

Реализация решения для платформы Web.

Ограничения целостности данных

- **Виды ограничений**

- на начальное значение атрибута (определяет значение атрибута по умолчанию)
- вычисляемый атрибут (используется для вычисления значения атрибута на основе значений других атрибутов)
- ограничение значений (определяет "валидационные" правила, такие как максимальное значение, минимальное значение и т.п.)

- **Способ указания**

- OWL аннотации к атрибуту (способ описания метаданных, предоставляемый языком OWL)

- **Использование ограничений**

- Ограничения автоматически проверяются системой при создании или редактировании информационного ресурса

Обеспечение минимальной функциональности манипулирования данными

- Постановка задачи в рамках Web платформы
 - Генерация web страниц предоставляющих базовые операции: просмотр, редактирование, поиск, удаление
- Решение:
 - Генерация проводится с использованием данных о визуализации и ограничениях, описанных в онтологии
 - Автоматически формируются на основании онтологии, шаблонов и данных страницы поиска, просмотра, редактирования и страница индекса

Реализация решения для платформы Web.

Итог

- Результат:
 - прототип информационной системы, с возможностью поиска, редактирования и просмотра любого ресурса, объявленного в онтологии
 - возможность использовать метаданные при написании произвольных пользовательских страниц
- Необходимые действия со стороны разработчика
 - описать структуру данных
 - определить ограничения целостности данных
 - задать соответствие между полями классов и шаблонами визуализации

Пример использования решения

- Онтология, описанная на языке OWL

```
<owl:Class rdf:about="News">
  <rdfs:label xml:lang="ru">Новость</rdfs:label>
</owl:Class>

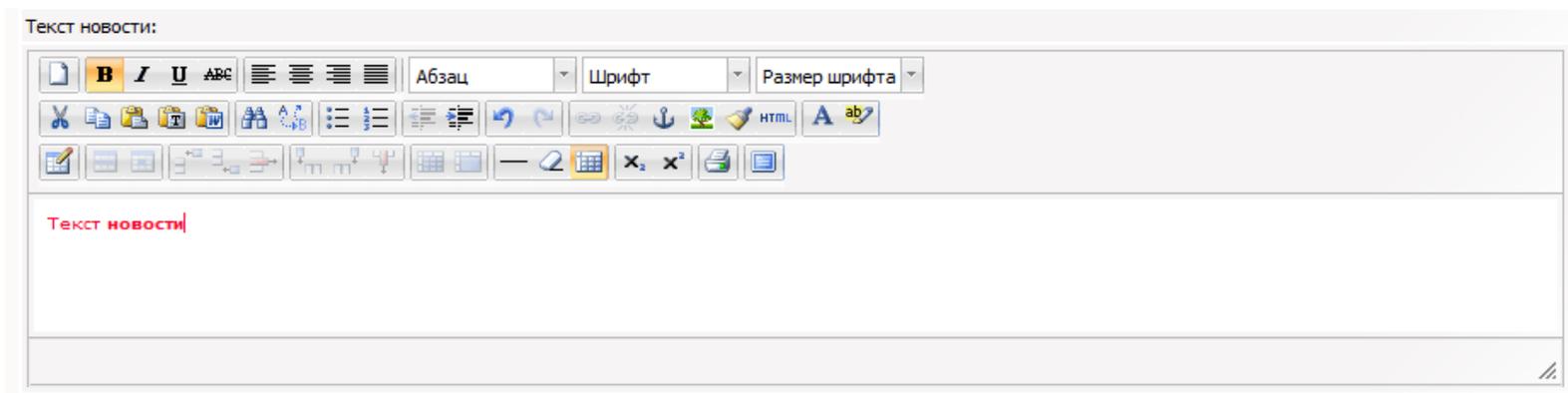
<owl:FunctionalProperty rdf:about="newsContent">
  <rdfs:label xml:lang="ru">Текст новости</rdfs:label>
  <rdfs:range rdf:resource="&xsd:string" />
  <rdfs:domain rdf:resource="News" />
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty" />

  <ui:customizeTemplate>
    <ui:type rdf:datatype="&xsd:string">edit</ui:type>
    <ui:templateName rdf:datatype="&xsd:string">text</ui:templateName>
    <ui:templateParam>
      <ui:name rdf:datatype="&xsd:string">type</ui:name>
      <ui:value rdf:datatype="&xsd:string">rich</ui:value>
    </ui:templateParam>
  </ui:customizeTemplate>
</owl:FunctionalProperty>
```

- В онтологии описан класс “News”, содержащий свойство “newsContent”, типа “String”. Также с помощью аннотации “ui:customizeTemplate” указано, что этому свойству, для типа отображения «на редактирование», соответствует шаблон “text” с параметром “type” = “rich” (т.е. свойство text должно выводиться пользователю в виде богатого текстового редактора).

Пример использования решения

- Визуальное отображение, описанного в онтологии класса новость.
 - Поле «Текст новости» выводится пользователю в виде богатого текстового редактора, т.к. в онтологии было указано значение “rich” для параметра “type”





Конец

Спасибо за внимание