

АВТОМАТИЗАЦИЯ РАЗРАБОТКИ ИНФОРМАЦИОННЫХ СИСТЕМ НА БАЗЕ OWL ОНТОЛОГИЙ

А.А. Бездушный

В работе рассматривается общая архитектура системы автоматизации разработки информационных систем. Система параметризуется описанием объектной схемы данных конкретной предметной области и легко адаптируется к её изменениям. Для описания схемы используется W3C стандарт на описание схем Интернет ресурсов - RDFS. Архитектура имеет многоуровневую модульную организацию, каждый уровень имеет собственные цели и абстракции. Система не имеет привязки к конкретной платформе или технологии разработки, а лишь определяет основные принципы организации информационного портала.

Аутентификация, фильтры доступа – блок отвечает за проверку прав доступа. На этом этапе происходит аутентификация пользователя. При поступлении запроса производится проверка на наличие прав доступа к запрошенному ресурсу. Дальнейшая обработка запроса производится только в случае успешного прохождения этого этапа. Для проведения проверки используется блок «Безопасность», описанный ниже, блок предоставляет программный интерфейс управления доступом. В частности деление пользователей на группы и определение привилегий пользователей.

Пользовательский интерфейс – блок формирует визуальное представление запрошенного ресурса и определяет обратную связь пользователя с сервером. Блок состоит из модулей – «Контроллер», «Визуализация доменных объектов», «View». «Контроллер» - осуществляет подготовку данных в соответствии с пользовательским запросом. Подготовленные данные передаются в один из модулей визуализации.

Визуализация доменных объектов - технология, направленная на создание пользовательского интерфейса объектов предметной области. Блок предоставляет способ формализации пользовательского интерфейса доменных объектов. В большинстве случаев визуальный интерфейс объекта предметной области определяется визуальным интерфейсом каждого его свойства и их взаимным расположением. Для формализации процесса визуализации вводятся понятия **шаблон отображения** и **каркас отображения**. *Шаблон отображения* определяет визуальный интерфейс конкретного свойства объекта, *каркас отображения* — компоновку визуальной среды пользователя. Таким образом, определив шаблоны отображения для каждого свойства и выбрав каркас, в котором эти шаблоны размещаются, мы полностью определяем визуальный интерфейс ресурса. За счет описанного разбиения достигается возможность декларативного, а не императивного создания пользовательского интерфейса доменных объектов. «View» - произвольная технология визуализации.

Системой определяется набор стандартных шаблонов, которые могут быть использованы при создании приложения. При необходимости можно создать новый шаблон. Созданные шаблоны помещаются в общий репозиторий шаблонов, благодаря чему их использование становится возможным в дальнейшем при создании других приложений.

Аналогично определен и набор стандартных каркасов, который также может быть расширен и использован не однократно.

Назовем объект, содержащий сведения о том, какие данные необходимо выводить, какие шаблоны для этого нужно применить и каким образом расположить их друг относительно друга – **объектом визуализации**.

Используя понятие объекта визуализации, мы получаем возможность описать визуальный интерфейс чисто декларативно, не прибегая к применению конкретных технологий. Такие, своего рода метаданные, могут быть описаны с помощью языка OWL. При использовании OWL информация о том, как визуально выглядит объект, становится пригодной для машинной обработки.

Компоненты внешнего доступа (адаптеры) – блок предоставляет функционал для интеграции со сторонними системами. В частности по протоколам SOAP, REST. Блок предоставляет интерфейс создания сервисов, а также автоматической генерации схем описания функциональности сервиса на формальных языках, таких как WSDL.

Ограничения на данные – блок осуществляет проверку введенных данных. При получении данных от пользователя ИС или от сторонних систем, данные автоматически проверяются на корректность и целостность. Системой определен набор стандартных ограничений которые можно использовать при создании приложения. При необходимости можно создать новое ограничение. Все ограничений хранятся в собственном репозитории, благодаря этому созданные ограничения в дальнейшем можно повторно использовать в других приложениях.

Прикладная логика приложения – блок, представляющий логику функционирования системы. Для автоматизации создания и моделирования прикладной логики, может быть использованы технологии MDA (Model driven architecture), BPMN (Business Process Model and Notation).

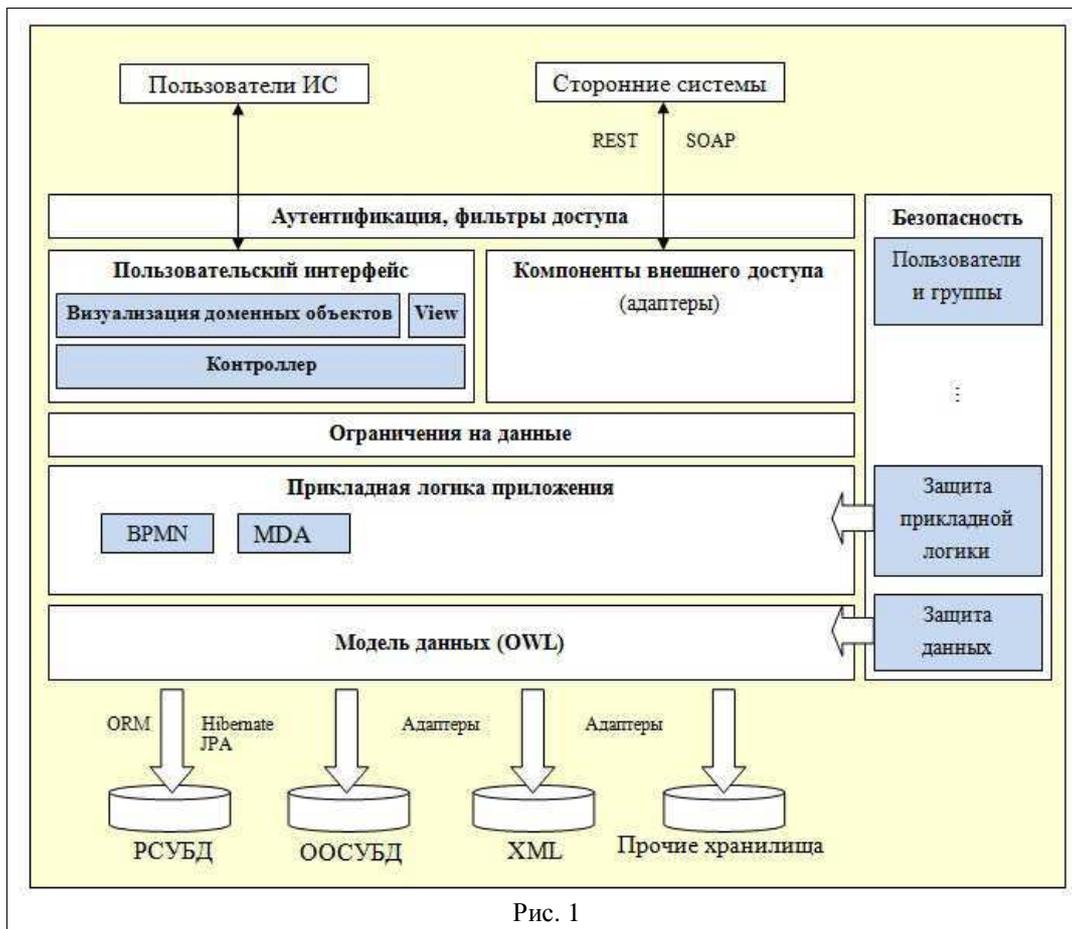


Рис. 1

Модель данных – блок, отвечающий за описание схемы данных предметной области и способы доступа к данным (адаптеры к хранилищам данных, таким как РСУБД, ОСУБД и др.). Схема данных описывается на языке OWL. Это предоставляет возможности обмена и машинного анализа данных. С помощью адаптеров, модель данных отображается в конкретное хранилище данных. В одном приложении данные могут получаться из нескольких источников одновременно.

Безопасность – блок обеспечивающий защиту данных. Блок состоит из набора модулей, обеспечивающих управление доступом к разным видам операций. Разделение блока на модули, позволяет выстроить многоуровневую защиту – на уровне конкретных классов, объектов или методов прикладной логики.

Модуль авторизации — предоставляет программный интерфейс управления доступом. Вводятся такие понятия как, пользователь, группа пользователей, привилегии пользователей.

Модуль защиты прикладной логики — отвечает за проверку прав доступа к компонентам прикладной логики приложения. Доступ может быть ограничен как к компоненте целиком так и к некоторым её функциям. Предоставляется интерфейс декларативного описания прав доступа на основе введенных выше понятий групп и привилегий.

Модуль защиты данных — отвечает за проверку прав доступа к объектам предметной области, за управление их правами доступа. Права доступа выражаются в виде списка разрешений - Access Control List. Каждый элемент списка – это тройка <субъект доступа, объект доступа, привилегия>. Субъект доступа - это пользователь или группа пользователей.

ЛИТЕРАТУРА:

1. А.А. Бездушный, А.К. Нестеренко, А.Н. Бездушный, В.А. Серебряков, Т.М. Сысов. Архитектура и технологии RDFS-среды разработки цифровых библиотек и Web-порталов <http://www.elbib.ru/index.phtml?page=elbib/rus/journal/2003/part4/BNSBS>
2. П.Н. Михеев, А.М. Меденников, А.Н. Бездушный : Проектирование информационных веб-порталов в ИСИР
3. P. Almedia MDA – Model Driven Architecture Improving Software Development Productivity in Large-Scale Enterprise Applications
4. Semantic Web Activity. <http://www.w3.org/2001/sw>
5. RDF Primer. W3C Working Draft. <http://www.w3.org/TR/rdf-primer>