

Таким образом, для решения задачи интеграции среды разработки Microsoft Visual Studio и системы электронного документооборота DIRECTUM использовалось COM-соединение с вызовом внешних объектов.

Список литературы

1. DIRECTUM. [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/Directum>
2. Особенности ASP.NET MVC 5. [Электронный ресурс]. – URL: <http://metanit.com/sharp/mvc5/1.1.php>

УДК 004

ИНТЕГРАЦИЯ ДАННЫХ НА ОСНОВЕ ТЕХНОЛОГИЙ SEMANTIC WEB И BIGDATA

Нгуен М.Х., Тузовский А.Ф.

Научный руководитель: Тузовский А.Ф., д.т.н., профессор каф. ОСУ, ИК ТПУ

*Национальный исследовательский Томский политехнический университет,
634050, Россия, г. Томск, пр. Ленина, 30
E-mail: hungnm.k52tnvlkt@gmail.com*

The possibility of using Hadoop Big Data platform for the implementation of a virtual system data integration based on Semantic Web technologies is described.

Ключевые слова: интеграция данных, Semantic Web, платформа Hadoop.

Key words: data integration, Semantic Web, platform Hadoop.

Под системой интеграции понимается информационные системы, позволяющие пользователям работать с набором источников разнотипных информации и данных как с единым целым: формировать запросы на получение необходимых данных, находящейся в различных источниках и получать результат в унифицированном виде. Каждый из источников может содержать некоторые структурированные (или полуструктурированные) данные и использовать различные модели данных, такие, как реляционная, нетрадиционная (NoSql) и семантическая модель. В общем случае, источники данных являются распределенными, т. е. расположенными на разных компьютерах, к которым есть доступ по компьютерной сети (по интранет или Интернет). Разные источники данных могут поддерживаться разными группами пользователей.

Существуют два основных подхода: виртуальная и материализованная интеграция. Виртуальная интеграция осуществляется с использованием технологии предметных посредников образующих промежуточный слой между пользователем и источниками данных. При материализованной интеграции предполагается создание хранилища данных (warehouse), которое периодически пополняется из интегрируемых источников данных. Каждый из этих подходов имеет свои достоинства и недостатки.

Общей проблемой классических систем интеграции является необходимость семантического (смыслового) согласования данных. Для решения таких проблем в последнее время

достаточно успешно используются технологии Semantic Web (TSW). TSW – это набор стандартов и методов, такие как RDF – формат данных в виде триплетов, OWL – язык описания семантических моделей (онтологий) данных и язык SPARQL – описание запросов к RDF данным [1].

Однако в последние десятилетия объемы данных фактически во всех областях деятельности растут экспоненциально. Для работы с такими большими объемами данных был сформирован новый подход BigData, включающий набор специальных технологий, таких как Hadoop, Spark, предоставляющих инструменты и методы обработки данных огромных объемов и значительного многообразия для получения воспринимаемых человеком результатов, эффективных в условиях непрерывного роста и распределения источников данных. При использовании Hadoop работа может выполняться с различными структурами данных (в том числе и RDF), объем которых может достигать нескольких миллиардов элементов [2].

В докладе предлагается новая архитектура интеграции данных, основанная на объединении возможностей технологий Big Data и существующих подходов виртуальной интеграции разнородных и распределенных данных больших объемов (рис. 1).

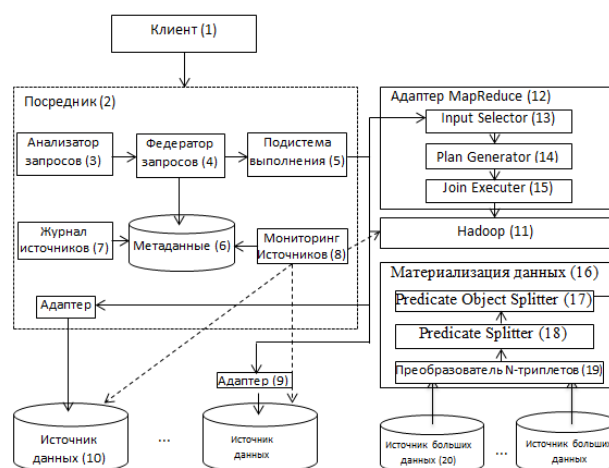


Рис. 1. Архитектура комбинированной виртуально-материализованной интеграции

Основными архитектурными принципами работы данной системы являются следующие: пользователи выполняют запросы к посреднику (2) с помощью клиента (1), т. е. передают на обработку глобальный SPARQL-запрос, грамматический анализатор (3) формирует канонический план запроса, который преобразуется и оптимизируется федератором (4) на основе информации в каталоге метаданных (6) (данный каталог периодически проверяет и обновляет свои статистические данные с помощью журнала (7) и мониторинга (8) источников). Разделенный на части, и оптимизированный глобальный план запроса состоит из нескольких локальных подпланов, которые далее отправляются и параллельно выполняются на источниках данных (10), после того результаты объединяются высокоуровневыми алгебраическими операциями подсистемы выполнения (5). Для каждого источника и его соответствующей модели данных имеется специальный адаптер (9), позволяющий преобразовать данные в RDF-формат. При выполнении подзапроса к Hadoop (11) используется MapReduce-адаптер (12), состоящий из три компонента: Input Selector (13) принимает SPARQL-запрос от посредника, Plan Generator (14) определяет, сколько Map-Reduce заданий необходимо для выполнения запроса и передает информацию в Join Executer (15), который отправляет задания в Hadoop на выполнение. Материализация данных (16) в Hadoop осуществляют-

ся с помощью 3 компонентов: NTriples Converter (19) преобразует исходные файлы из источников больших данных (20) в формате N-Триплетов (n3), Predicate Splitter (18) принимает преобразованные данные, разделит их в соответствии с предикатами и хранит в предикатных файлах, Predicate Object Splitter (17) расщепляет такие предикатные файлы на файлы меньшего размера и отправляет в Hadoop.

Таким образом, в комбинированной архитектуре интеграция становится двухслойной: материализация больших данных осуществляются путем использования Hadoop и виртуальная семантическая интеграция на уровне предметных посредников. Это дает возможность распределенного хранения, преобразования и интеграции больших разнотипизированных данных, а также унифицированный взгляд на данные через семантическую модель.

Список литературы

1. Langegger A.A. Flexible architecture for virtual information integration based on semantic web concepts, Ph.D. dissertation, Johannes Kepler University Linz, 2010.
2. Husain M.F., Doshi P., Thuraisingham B.M. Storage and Retrieval of Large RDF Graph Using Hadoop and MapReduce, CloudCom. 2009. pp. 680–686.

УДК 004

ПРОЕКТИРОВАНИЕ ИГРОВОГО 2D ПРИЛОЖЕНИЯ

Потапкина С.В., Коцин Д.О.

Научный руководитель: Вичугова А.А.

Национальный Исследовательский Томский политехнический университет,

634050, Россия, г. Томск, пр. Ленина, 30

E-mail: w_snsam@yahoo.com

The article tells about software development of 2D games.

В статье описан начальный этап проектирования игрового приложения: представлен сюжет игры и основные технологии реализации.

Key words: *unity, 2D, visual studio, c#, game, indie, graphics editor.*

Ключевые слова: *unity, visual studio, c#, 2D, игра, инди, графический редактор.*

Введение

В настоящее время игровая индустрия стремительно развивается: каждый день создаются и публикуются игры, которые впоследствии находят свою аудиторию среди разных возрастных групп. При этом, количество игроков ежедневно увеличивается за счёт приобретения ими компьютеров, приставок, смартфонов.

Предметом рассмотрения данной статьи является процесс проектирования инди-игры. Инди-игра (англ. Indie game, от англ. independent video game – «независимая видеоигра») – это видеоигра, которая создана независимым разработчиком или коллективом разработчиков, без финансовой поддержки от издателя игр. Главные особенности независимой игры заключаются в реализации новой идеи, игровой механики, уникального стиля, а вырученные за