

Интеграция распределенных данных на основе технологий Semantic Web и рабочих процессов

Нестеренко А.К.
Бездушный А.Н.

Сысоев Т.М.

Бездушный А.А.
Серебряков В.А.

Вычислительный центр им. А.А. Дородницына РАН

alexn@ccas.ru, tim@ccas.ru,
alix@7ka.mipt.ru, bezdushn@ccas.ru

Аннотация

В данной статье рассматривается один из подходов к осуществлению задач интеграции распределенной информации посредством выполнения открытых запросов к внешним системам под контролем управляемых потоков работ. Рассматриваются возникающие задачи и направления дальнейшего развития подхода.

1 Введение

В наш век, когда происходит бурное развитие научных и исследовательских работ по различным направлениям, одной из важнейших задач становится интеграция накопленных знаний и данных. При этом любой ученый или исследователь должен иметь возможность не только обратиться к изолированной информационной системе для получения узкоспециализированной информации по интересующему его предмету, но и получать агрегированные по многим источникам данные, которые помогли бы ему получить представление о конкретных объектах в целом. Все возрастающая потребность в получении «обобщенного» взгляда на информационные ресурсы различных систем привела к формированию механизмов интеграции данных, которые ориентируются на эффективную организацию доступа к внешним, гетерогенным источникам данных посредством единого интерфейса в рамках единой модели данных. При этом сам метод агрегирования информации должен быть настолько прозрачным для конечного пользователя, чтобы в его глазах процесс получения информации ничем не отличался от обращений к единому хранилищу данных. Процесс интеграции информации включает в себя этапы формирования и рассылки запросов внешним системам, преобразования запросов, их результатов

между представлениями частных и канонической моделей и схем, сбора результатов и их объединения в соответствие с каноническими моделью и схемами данных, организации локального хранения данных о внешних системах и других рабочих ресурсах, кэширования промежуточных результатов, управление потоком запросов и агрегирования ответов и т.п.. Задачи интеграции данных, а особенно приложений и сервисов, тесно переплетаются с задачами организации и поддержки рабочих процессов, которые, будучи управляющими механизмами, позволяют реализовать управляемую логическую последовательность работ по сбору и агрегированию информации в ответ на пользовательский запрос. В статье рассматривается некоторый подход к интеграции распределенных данных, базирующийся на технологиях объектных репозиторий, Web-сервисов, открытых запросов и рабочих процессов управления формированием результирующей информации.

2 Существующие решения в области интеграции распределенных данных

На сегодняшний день основными по объему источниками структурированных данных выступают реляционные базы данных, хотя это могут быть и файловые системы, и XML базы данных, расширяющие масштабы своего применения, и другие типы источников информации. Вне зависимости от выбранного метода хранения данных первая проблема интеграции гетерогенных данных, с которой приходится сталкиваться при формировании репозитория информационных ресурсов, это разнообразие моделей и схем данных, низкий уровень их абстракции, малая адекватность отражения семантики предметной области, для качественного решения которой необходимо работать с данными, соответствующими некоторой существенно семантически более богатой модели данных. Например, хорошим решением может быть переход к некоторой объектно-ориентированной модели данных, которые по многим параметрам

близки к семантическим моделям, где ключевой единицей является сложно структурированный информационный объект, поддерживающий различные атрибуты, участвующий в различных ассоциациях с другими объектами. Поскольку нам необходимо осуществлять интеграцию разнообразных источников информации, где существенную роль играют реляционные источники, то существенное значение принимает поддержка интерфейса, обеспечивающего автоматическое отображение конструкций семантической модели данных в нижележащую, например, реляционную модель данных, и наоборот. Для описания объектно-ориентированных моделей данных можно воспользоваться ряд языков описания объектных схем данных, например:

- ODL - стандарта ODMG[7] объектно-ориентированных БД;
- RDFS (Resource Definition Framework Schema[4]) – W3C стандарт позволяет описывать схемы классов и их свойств, с учетом их наследования, ограничений;
- OWL (Web Ontology Language[8]) – специализация RDFS, ориентированная на описание предметных онтологий.

Переход от реляционной модели данных к объектно-ориентированной является необходимым этапом в построении открытого информационного хранилища. Объектно-ориентированная модель позволяет:

- повысить уровень абстракции модели предметной области;
- удобно выделить канонические схемы данных, представляющие собой пересечение экспортных схем данных, соответствующих различным предметным областям, строить унифицированные объектные запросы на доступ к распределенным данным с последующим агрегированием результатов запросов в соответствие с каноническими схемами;
- в случае RDFS и OWL – еще иметь унифицированный формат представления данных, обеспечивающий неплохую синтаксическую и семантическую интероперабельность.

В вопросе интеграции (точнее, технической интероперабельности) распределенных репозиториях данных все большую силу набирает технология Web-сервисов[12], как средства предоставления унифицированного, платформенно-независимого интерфейса для удаленного доступа к информационным ресурсам. В данном контексте Web-сервис выступает в роли автономного приложения, которое предоставляет средства доступа к информации внешним клиентам через набор предоставляемых им услуг. Технология Web-сервисов базируется на таких открытых XML-стандартах, как:

- SOAP (Simple Object Access Protocol[6]) – XML-протокол для удаленного вызова методов Web-сервисов;
- UDDI(Universal Description, Discovery and Integration[9]) – описывает модель данных, предназначенную для каталогизации и обнаружения услуг, предоставляемых Web-сервисами;
- WSDL (Web Services Description Language[10]) – язык описания интерфейсов Web-сервисов.

Формирующиеся дополнения к ним, например, WS-Coordination/WS-Transaction (транзакции), WS-Security (безопасность), WS-Routing (маршрутизации сообщений) и т.д., призваны расширить возможности этой платформы в удовлетворении требований задач интеграции приложений. В рамках инициативы WS-I разрабатываются примеры прикладных решений, предложения и дополнительные требования, призванные гарантировать совместимость решений разных поставщиков. Это сулит широкие возможности по интеграции различных информационных систем в рамках единого согласованного набора спецификаций.

Во многих случаях интеграция информационных ресурсов требует комбинирования обращений более чем к одному Web-сервису для реализации пользовательского запроса. Таким образом, Web-сервисы должны иметь возможность поддерживать взаимодействие с другими приложениями в дополнение к стандартным процедурам обработки данных. Более того, процесс предоставления агрегированной распределенной информации может включать в себя разбиение на набор взаимосвязанных этапов обработки данных, взаимодействие ряда Web-сервисов, вмешательство людей в процесс обработки пользовательских запросов и другие элементы прикладной логики. Поэтому процесс сбора и интеграции гетерогенных данных может представлять собой логически-сложную композицию обращений к хранилищам информационных сущностей посредством интерфейсов Web-сервисов – определять автоматизированный поток обработки данных.

Для описания композиций Web-сервисов на данный момент различными ассоциациями предлагается ряд стандартов. Среди них можно отметить следующие языки описания автоматизированных потоков работ, участниками которых являются Web-сервисы:

- WSFL (Web Services Flow Language[11]) – позволяет определять композиции Web-сервисов в виде графовой модели рабочего процесса;
- BPML (Business Process Modeling Language[1]) – определяет блочную модель композиции Web-сервисов;
- BPEL4WS (Business Process Execution Language For Web-Services[2]) – представляет собой гибрид блочной и графовой моделей описания взаимодействий Web-сервисов.

Эти языки позволяют описывать композиции Web-сервисов, что позволяет определять сложные, распределенные процессы по извлечению, обработке и интеграции информации.

Итак, мы можем выделить метод осуществления процесса сбора и интеграции распределенных данных, который базируется на трех технологиях:

- объектные репозитории данных, соответствующие некоторым предметным областям;
- механизм Web-сервисов, как средство построения внешних интерфейсов к таким репозиториям;
- аппарат рабочих процессов, как средство управления обработкой и интеграции информационных потоков.

3 Использование механизма выполнения открытых запросов для решения задачи интеграции данных распределенных репозиториях

На основе анализа подходов к решению задачи интеграции распределенных данных, была разработана концепция так называемых «открытых запросов» к внешним системам и предложена система, обеспечивающая их исполнение. Она обеспечивает методическую и программную поддержку формирования автоматизированного программного решения по осуществлению открытых запросов к хранилищам внешних систем на базе унифицированного языка запросов (вариации языка ODMG OQL) и унифицированной формы представления удовлетворяющих запросам данных (W3C RDF/XML).

3.1 Формирование объектных репозиториях существующих информационных систем

Понятие «интеграция распределенных данных» подразумевает, как правило, интеграцию информационных ресурсов, которые расположены в уже существующих распределенных репозиториях. В настоящее время большая часть информационных хранилищ представлена реляционными базами данных. Поэтому первая задача, возникающая на пути решения проблемы семантически обоснованной интеграции информационных ресурсов – это представление данных, описанных реляционной моделью, семантически более богатым способом. Таким образом, необходимо наличие механизмов, позволяющих выделить из реляционной модели данных объектную модель и реализовать адаптер для работы с данными существующего хранилища информационных ресурсов через объектные интерфейсы доступа (такие как, например, ODMG API или ODMG OQL).

Была использована методика, которая опиралась на реинжиниринг реляционных схем данных существующих реляционных хранилищ данных,

создание соответствующих объектных схем данных и возможности программного комплекса, базирующегося на Java-технологиях, которые позволяют сформировать «объектную» надстройку над имеющимся реляционным хранилищем информационных ресурсов для того, чтобы работать с его данными посредством технологий Semantic Web в рамках канонической RDFS-модели данных [3]. Реализованные в рамках работы средства включают:

- Автоматизированную пользовательскую среду подготовки унифицированного, ориентированного на поддержку семантической интероперабельности, описания схем данных (RDFS), предоставляемых хранилищами внешних систем, включающую:
 - Реинжиниринг (восстановление) исходной реляционной схемы БД учетом специфики провайдеров РСУБД.
 - Преобразование реляционной схемы данных в объектную схему данных.
 - Преобразование объектной схемы данных в RDFS схему данных.
- Формирование репозитория хранимых RDFS-объектов над реляционной БД внешней системы, включающее:
 - Формирование объектной и RDF прослойки над реляционной БД, параметризованных RDFS-схемой данных репозитория. Соответствующее обеспечение объектно-реляционного и RDF-объектно отображений данных, параметризованных декларативными описаниями соответствующих отображений и схем данных.
 - Поддержку диалекта объектного языка запросов ODMG OQL, обеспечивающую трансляцию OQL запросов в SQL запросы с учетом специфики провайдеров РСУБД.
- Представление данных хранилищ внешних систем, выбираемых открытыми запросами, в унифицированной W3C RDF/XML форме, обеспечивающей возможность осуществления семантической интероперабельности данных.
- Предоставление удаленного обращения с открытыми запросами к репозиторию хранимых объектов на основе Web-сервисов, поддерживающих взаимодействие по протоколу SOAP.

Основные этапы формирования объектного репозитория открытой системы схематически представлены на **рис. 1**.

Для выделения объектной схемы реляционных баз данных внешних систем в рамках разработанной методики необходимо выполнить следующую последовательность действий:

- 1) **Формирование ER-схемы для БД целевой системы.** На первом этапе необходимо получить схему существующей реляционной

базы данных для того, чтобы впоследствии преобразовать ее к объектной схеме, внося дополнительные семантические наполнение и структуризацию. Выделение ER-схемы существующей БД целевой системы можно выполнить следующими программными средствами:

- **MS Visio 2000/2002/2003** (позволяет построить системную ER-схему БД в ER-нотации);
- **IBM RROSE 2000/2002/2003** (позволяет с помощью модуля *Data Modeller* сформировать системную ER-схему целевой БД).

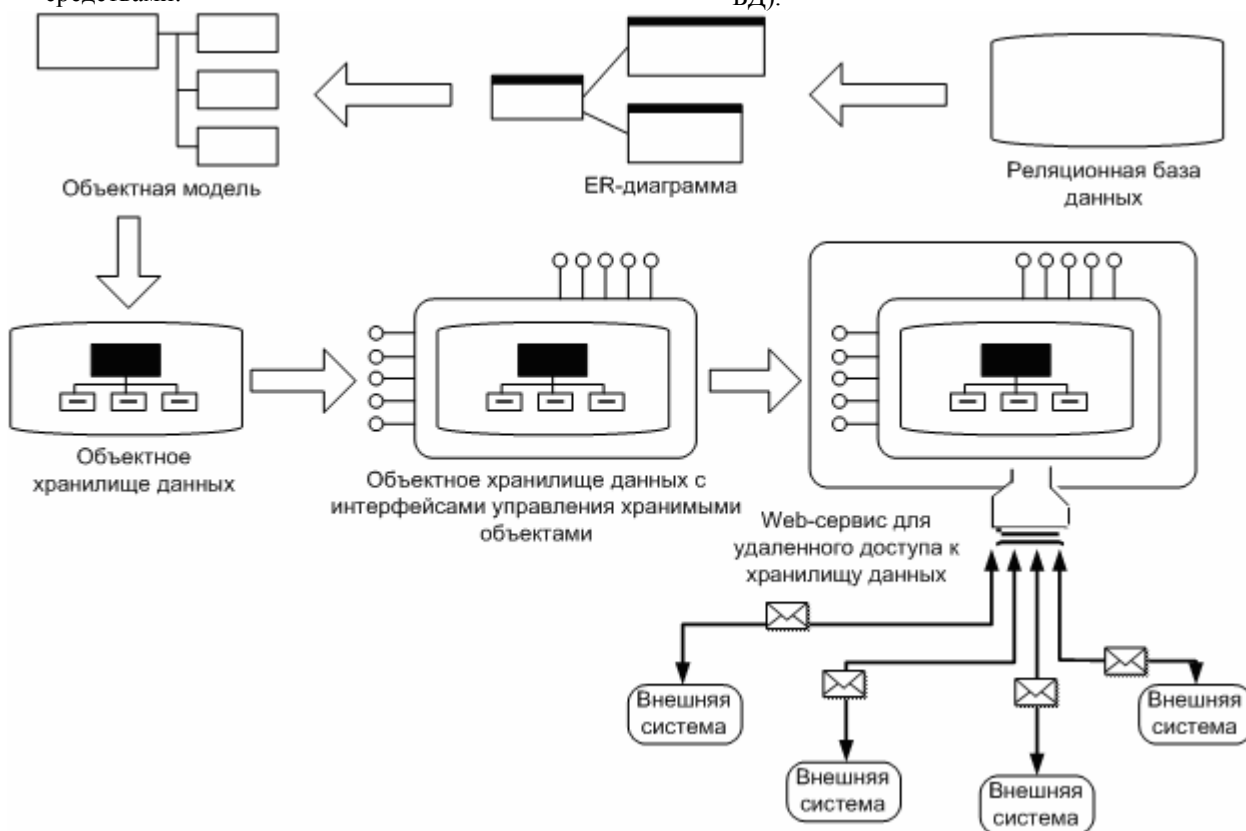


Рисунок 1. Методика построения объектных репозиториях открытых систем

2) **Формирование UML-диаграммы классов по ER-схеме целевой системы.** Второй этап в построении объектного репозитория над реляционной базой данных, это преобразование полученной ER-схемы данных к первому приближению RDFS-модели информационных ресурсов Semantic Web. В качестве этого первого приближения удобно использовать UML-диаграмму классов. Формирование UML-диаграмм классов по ER-схемам можно выполнить следующими программными средствами:

- **MS Visio 2000/2002/2003** (не умеет преобразовывать ER-схемы в UML-диаграммы классов, ввиду чего требуемое преобразование необходимо выполнить «руками», имея в редакторе две этих схемы);
- **Poseidon for UML** (не умеет преобразовывать ER-схемы в UML-диаграммы классов, ввиду чего требуемое преобразование необходимо выполнить «руками», имея в редакторе две этих схемы);

• **IBM RROSE 2000/2002/2003** (представляет ER-схему в UML-нотации по собственной методике).

3) **Запись UML-диаграммы классов в OMG XMI формате.** На следующем этапе нам необходимо представить полученную UML-диаграмму классов в некоторой промежуточной, схемонезависимой форме для последующего преобразования к модели данных RDFS. В качестве такого промежуточного представления в данной методике выбран OMG XMI формат представления объектных схем. Запись UML-диаграмм классов в OMG XMI форме, необходимой для получения первого варианта RDFS-схемы, можно выполнить следующими программными средствами:

- **MS Visio 2002/2003** (имеется addon, выгружающий UML-диаграммы классов в OMG XMI формате);
- **Poseidon for UML CE** (в рамках архивов zargo сохраняет UML-диаграммы в OMG XMI формате);

- **IBM RROSE 2002/2003** (имеется plugin, записывающий UML-диаграммы классов OMG XMI формате).
- 4) **Преобразование UML-диаграмм классов в OMG XMI форме в RDFS-схему.** На данном этапе подготовлены все необходимые входные артефакты для построения первого варианта RDFS-схемы, описывающей схему данных объектной надстройки над реляционной базой данных. Преобразование UML-диаграмм классов в OMG XMI форме в первый вариант RDFS схемы и последующее ее преобразование выполняется с помощью редактора онтологий **Protégé-2000**, обеспечивающего поддержку RDF и RDF Schema.
- 5) **Формирование прикладной RDFS-схемы.** После выделения первого приближения RDFS-модели данных объектного репозитория необходимо выполнить доработку полученного первого варианта схемы до семантически более корректной формы. На данном этапе предполагается:
- Доработка RDFS-схемы данных: выделение введение дополнительной иерархии классов и их свойств;
 - Введение системных классов технологической платформы, необходимых адаптеру объектного репозитория;
- 6) **Формирование RDFS схемы, согласованной с совокупностью канонических RDFS подсхем.** На данном этапе имеется выделенная RDFS-схема объектной надстройки над реляционным хранилищем данных. Для возможности интеграции информационных ресурсов репозитория различных внешних систем, описанных подобными схемами, необходимо выделить из них канонические (общие) подсхемы, в рамках которых будут формироваться объектные запросы на доступ к информационным ресурсам и осуществляться интеграция полученных от различных внешних систем ответов. В свете этого на данном этапе пространство имен прикладной RDFS-схемы разбивается на следующие три:
- Пространство имен **common** – каноническая RDFS-подсхема общих классов, свойств, в соответствии с которыми могут формироваться объектные запросы;
 - Пространство имен **external** – каноническая RDFS-подсхема общих прикладных классов, свойств, в соответствии с которыми пользователю могут возвращаться данные прикладной системы;
 - Пространство имен **external_own** – RDFS-подсхема общих прикладных классов, свойств, которые поддерживаются репозиторием, но недоступны объектным запросам.
- 7) **Реализация адаптера объектного репозитория – поддержка прикладных RDFS-схем, согласованных с совокупностью**

канонических RDFS-схем. На данном этапе сформированное полноценное описание объектной схемы данных репозитория используется как входной параметр для реализованного адаптера объектного репозитория, который позволяет [15]:

- Осуществить объектно-реляционное отображение полученной объектной схемы данных на реляционную схему существующей реляционной БД;
- Выполнять объектные OQL-запросы к репозиторию, согласованные с канонической RDFS-схемы общих классов;
- Представлять результаты OQL-запросов к репозиторию в унифицированном RDF/XML формате;
- Предоставить Web-сервис для выполнения OQL-запросов к сформированному объектному репозиторию и получения RDF/XML ответов.

Таким образом, разработанный метод построения объектных репозиториев над имеющимися информационными хранилищами данных в довольно большой степени решает первую часть проблемы интеграции данных, обусловленную необходимостью повышения уровня семантического представления данных распределенных информационных систем.

3.2 Интеграция информационных ресурсов объектных хранилищ данных посредством рабочих процессов

Для управления интеграцией данных, поступающих из распределенных объектных репозиториев, методика формирования которых описана выше, были разработаны средства поддержки композиций Web-сервисов репозиториев внешних систем, которые базируются на стандарте BPEL4WS (Business Process Execution Language For Web-Services), разработанном совместно компаниями IBM и Microsoft. Данный стандарт был выбран в результате анализа и сравнения существующих подходов к описанию автоматизированных потоков работ ввиду того, что он обладает достоинствами обоих подходов (графового и блочного) и предоставляет гибкие средства для описания взаимодействий Web-сервисов[5]. Кроме того, он рекомендован к применению совместно с другими последними стандартами в области рабочих процессов, такими как WS-Transaction[14] (протоколы координации транзакций Web-сервисов), WS-Security[13] (безопасный протокол передачи данных в рамках процесса, гарантирующий их целостность) и т.д. Таким образом, Workflow-процесс, описанный на языке BPEL4WS, позволяет реализовать всю логику обработки пользовательских запросов, управления взаимодействием с распределенными репозиториями, интеграции поступающих данных и

формирования агрегированного ответа пользователю.

В рамках описываемых работ для координации потока информации, поступающей из репозиториев внешних информационных систем, была разработана среда интерпретатора рабочих процессов, описанных с помощью языка BPEL4WS. Данное решение основано на Java-технологиях [15 - 17] и реализует следующий набор функций:

- 1) Поддержка реестра рабочих процессов, описанных на языке BPEL и средств для:
 - Регистрации в реестре новых описаний рабочих процессов;
 - Обновления и удаления имеющейся информации;
 - Получения информации о списке зарегистрированных в реестре процессов;
- 2) Внешнее представление рабочих процессов посредством отдельных WEB-сервисов с предопределенным интерфейсом для возможности конструирования композиций рабочих процессов;
- 3) Синхронное и асинхронное взаимодействие с участниками процесса;
- 4) Поддержка основных управляющих конструкций языка BPEL:
 - элементарные операции;
 - условные и циклические конструкции;
 - динамический вызов операций внешних WEB-сервисов;
 - обработчики системных событий и исключений;
 - последовательное и параллельное исполнение заданий процесса;
 - трансформация оперативных данных процесса от одного формата к другому;
- 5) Управление множественными экземплярами процессов и маршрутизация между ними поступающих в систему внешних сообщений.

Сценарий применения координирующих рабочих процессов для сбора и агрегирования полученной информации в ответ на пользовательский запрос разбивается на следующую последовательность действий:

- 1) Обработка пользовательского внешнего запроса на получение агрегированной информации из нескольких репозиториев внешних систем;
- 2) Трансформация этого запроса к форматам объектных OQL-запросов, воспринимаемых этими репозиториями и последовательный (или параллельный с последующей синхронизацией) опрос этих репозиториев с обработкой исключительных ситуаций (в частности, временных ограничений) и динамическим изменением списка опрашиваемых систем в случае возникновения ошибок;
- 3) Возможная предварительная трансформация и обработка поступивших из репозиториев внешних систем данных;

- 4) Агрегирование результатов опроса различных репозиториев. Данное агрегирование базируется на том факте, что возвращаемая репозиториями информация соответствует набору канонических RDFS-схем данных, что позволяет выполнить семантическое объединение распределенной информации об одном и том же ресурсе, уникально идентифицируемом значениями некоторого набора атрибутов канонической модели. Например, если информация о ФИО персоны попадает в каноническую RDFS-схему данных, описывающую ресурс «персона», то при семантически правильном построении репозиториев внешних систем, информация о персоне, поступающая из них, помимо специализированных данных будет в обязательном порядке содержать поля канонической схемы данных (ФИО). Если выбрать этот атрибут канонической модели как идентифицирующий данный ресурс, то может быть выявлена принадлежность информации из различных внешних систем одной и той же информационной сущности с последующим ее агрегированием.

Описанный механизм интеграции данных посредством координирующих рабочих процессов является одним из возможных применений технологии открытых запросов для интеграции распределенной информации.

На **рис. 2** схематически представлен процесс взаимодействия компонентов подсистемы исполнения открытых запросов:

Таким образом, выбор и реализация гибких декларативных средств по описанию и исполнению динамических сценариев взаимодействия участников процесса, базирующихся на BPEL4WS описаниях рабочих процессов для сбора и агрегирования информации, дает широкие возможности для дальнейшего развития алгоритмов семантически обоснованной интеграции распределенных данных.

3.3 Средства визуального моделирования рабочих процессов по интеграции данных

Для эффективного построения, внедрения и модификации BPEL4WS описаний координирующих потоков работ необходимо наличие среды разработки, предоставляющей средства визуального конструирования спецификаций рабочих процессов. Был проведен сравнительный анализ существующих подходов к построению средств визуального моделирования графовых и блочных описаний потоков работ. В результате анализа были выделены следующие два широко используемые подхода к решению данной задачи:

- 1) **Иерархическое редактирование описаний рабочих процессов.** Данный подход позволяет построить унифицированное средство моделирования рабочих процессов, так как

XML-описание потоков работ всегда можно представить иерархической визуальной моделью, иерархия уровней которой отражает иерархию уровней элементов XML-описания. Между тем реализованные по данной методике визуальные средства не обладают достаточной наглядностью и ориентированы скорее на

искусственных разработчиков, чем на достаточно широкий круг пользователей;

- 2) **Двумерный графический редактор описаний рабочих процессов.** Визуальные средства, основанные на этом подходе, обладают достаточной наглядностью и функционалом. Они позволяют формировать новые сценарии

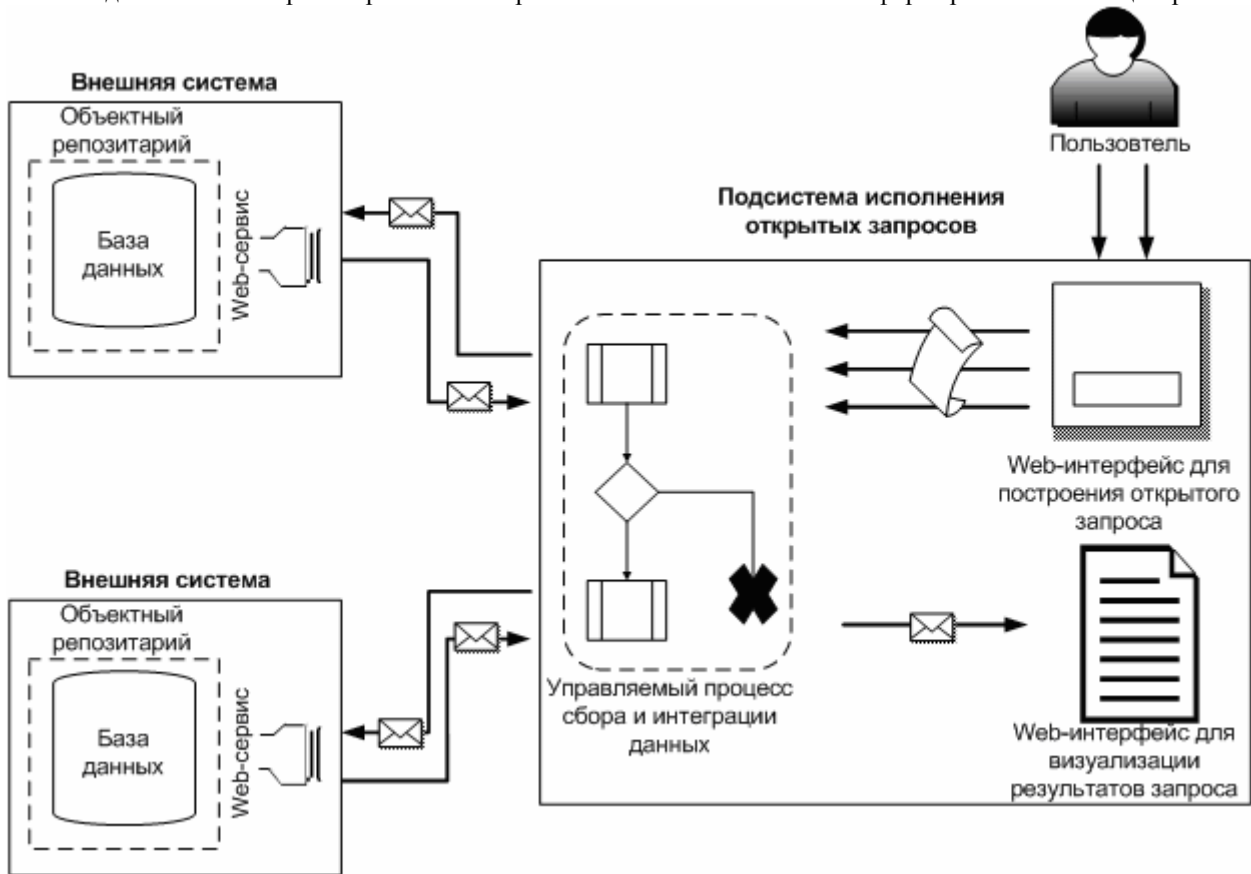


Рисунок 2. Процесс взаимодействия компонентов подсистемы исполнения открытых запросов

взаимодействия участников интеграции распределенных данных, как профессиональным разработчикам, так и менее искушенным в плане знания синтаксиса блочной структуры описания рабочих процессов аналитикам и экспертам.

Для разработки описаний автоматизированных потоков работ были реализованы визуальные средства редактирования древовидной структуры описания рабочего процесса на языке BPEL4WS, предоставляющие оба указанных способа редактирования потоков работ, которые позволяют:

- загружать/выгружать описания автоматизированных потоков работ;
- представлять описание процесса в виде иерархической и двумерной структуры его элементов;
- выполнять редактирование структуры описания, атрибутов его отдельных узлов;
- осуществлять расширенные синтаксические и семантические проверки описаний рабочих

процессов, позволяющие выявить статические ошибки до начала их функционирования;

- представлять рабочие процессы на разных уровнях выразительности, для разных ролевых частей формирования процессов, например, архитектор и разработчик.

С архитектурной точки зрения редактор описаний рабочих процессов поддерживает модульную структуру. К основным архитектурным блокам относятся:

- Модуль загрузки/выгрузки BPEL-документов;
- Модуль синтаксической и структурной верификации BPEL-описаний;
- Модуль редактирования атрибутов элементов BPEL-описания (набор XML-декларируемых форм);
- Модуль управления правами доступа к элементу описания и пользовательскими ролями (с расширенной поддержкой XML-декларации пользовательских ролей);

- Модуль управления локализацией пользовательских интерфейсов.

Для определения различных типов визуального представления BPEL-описаний и соответствующих средств редактирования его структуры редактор поддерживает понятие визуальных plugin-ов, каждый из которых может:

- Определять окно визуального редактора;
- Добавлять дополнительную палитру элементов и контекстное меню;
- Представлять свой локализационный ресурс;
- Назначать свою политику доступа к элементам описания для пользовательских ролей;
- Пользоваться услугами остальных модулей через стандартные интерфейсы.

Редактор поддерживает XML-регистрацию новых подключаемых plugin-ов визуальных редакторов и их автоматическое встраивание в пользовательский интерфейс. В данной версии визуального редактора BPEL-описаний потоков работ два реализовано два визуальных plugin-a:

- Иерархический (древовидный) редактор;
- Двумерный редактор описаний рабочих процессов.

В планах дальнейших работ по развитию средств визуального моделирования рабочих процессов среди наиболее важных задач стоит интеграция визуальных средств редактирования с интерпретатором описаний рабочих процессов для поддержки визуальной отладки и тестирования BPEL4WS процессов.

4 Дальнейшие шаги по решению задачи интеграции распределенных данных

Среди дальнейших работ по решению задачи интеграции информационных ресурсов с использованием технологии открытых запросов можно выделить следующие наиболее важные направления:

- разработка механизмов каталогизации информации о репозиториях внешних систем на базе интерфейсов UDDI, но с повышением уровня семантического описания для возможности динамического определения необходимых интегрирующему процессу источников информации для организации запросов – дальнейший шаг в сторону построения Semantic Web;
- внесение элементов семантического описания в декларацию потоков работ по сбору и интеграции данных для организации эффективного динамического поиска и выбора необходимых каталогизируемых источников данных для взаимодействия;

- развитие механизмов агрегирования ответов объектных репозиториях внешних систем на базе канонических RDFS-моделей данных;
- интеграция управляемых потоков работ по обработке данных с безопасными протоколами передачи информации, поддерживающими транзакционность выполнения этапов процесса, проверку целостности передаваемых данных, управление политикой безопасности данных при многопользовательском режиме работы;
- дальнейшее развитие средств моделирования описаний управляемых рабочих процессов, внедрение средств синтаксической и структурной статической верификации описаний потоков работ, элементов динамической валидации;
- интеграция визуального редактора описаний рабочих процессов с интерпретатором BPEL4WS рабочих процессов в плане разработки визуальных средств отладки и тестирования рабочих процессов методом «контрольных точек» (сравнение в выбранных узлах процесса текущего «разреза» оперативных данных с одним из эталонных для данной «контрольной точки»);
- дальнейшая автоматизация процесса динамического построения пользовательских интерфейсов для конструирования запросов на получение интегрированных данных в соответствие с каноническими объектными схемами предметных онтологий, участвующих в формировании интегрированного ответа.

5 Заключение

Подход к задаче интеграции данных посредством применения технологий объектных репозиториях, концепций Semantic Web, Web-сервисов и управляемых рабочих процессов по обработке информационных потоков является новым и перспективным решением, так как он позволяет:

- наделить хранилище информационных ресурсов большей семантической выразительностью;
- организовать эффективный доступ к информационным системам через унифицированный интерфейс;
- организовать сложные управляемые процедуры по извлечению и интеграции распределенных данных посредством использования декларируемых рабочих процессов, которые могут:
 - управлять параллельными заданиями по обработке данных;
 - заниматься автоматическим преобразованием данных на промежуточных этапах;
 - обеспечивать безопасность и целостность потоков данных;
 - управлять обработкой исключительных ситуаций;

- привлекать человеческие ресурсы на этапах, когда вопросы интеграции информационных ресурсов не могут быть решены автоматически;
- управлять динамическим выбором необходимых источников информации при соответствующем уровне семантического описания.

Интеграция информации распределенных внешних репозиториях требует наличия регламентированных, корректных с семантической точки зрения, динамически гибких и функциональных механизмов по решению задачи управления и координации информационных потоков. Именно поэтому применение объектных, Semantic Web и Workflow технологий дает более всесторонним решение этой сложной задачи.

Литература

- [1] BPML working draft March 25, 2002
<http://www.bpml.org/>,
<http://xml.coverpages.org/bpml.html>
- [2] Business Process Execution Language for Web Services Version 1.1
<http://www-106.ibm.com/developerworks/library/ws-bpel/>
- [3] Integrating Applications on the Semantic Web
<http://www.w3.org/2002/07/swint>
- [4] RDF Vocabulary Description Language 1.0: RDF Schema, W3C Working Draft 23 January 2003
<http://www.w3.org/TR/rdf-schema/>
- [5] Robert Shapiro, "A Comparison of XPD, BPML and BPEL4WS"
<http://xml.coverpages.org/Shapiro-XPDL.pdf>
- [6] SOAP Version 1.2 Part 1: Messaging Framework
<http://www.w3.org/TR/2003/REC-soap12-part1-20030624/>
- [7] The Object Database Standard: ODMG-93, Release 1.1, R. G. G. Cattell, Editor, Morgan Kaufmann Publishers, Inc., San Mateo, CA (1994).
- [8] The OWL Web Ontology Language
<http://www.w3.org/TR/2004/REC-owl-features-20040210/>
- [9] UDDI Version 3.0
<http://uddi.org/pubs/uddi-v3.00-published-20020719.htm>
- [10] Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language
<http://www.w3.org/TR/2004/WD-wsdl20-20040326/>
- [11] Web Services Flow Language
<http://www-3.ibm.com/software/solutions/WebServices/pdf/WSFL.pdf>, <http://xml.coverpages.org/wsfl.html>,
<http://www.ebpml.org/wsfl.htm>
- [12] Web Services for Business Process Design
http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/, <http://xml.coverpages.org/xlang.html>
- [13] Web Services Security (WS-Security)
<http://www-106.ibm.com/developerworks/WebServices/library/ws-secure/>
- [14] Web Services Transaction (WS-Transaction)
<http://www-106.ibm.com/developerworks/WebServices/library/ws-transpec/>
- [15] Бездушный А.А., Бездушный А.Н., Нестеренко А.К., Серебряков В.А., Сысоев Т.М. Архитектура RDFS-системы. Практика использования открытых стандартов и технологий Semantic Web в системе ИСИР. // Сборник докладов Пятой Всероссийской конференции "Электронные библиотеки: перспективные методы и технологии, электронные коллекции", Санкт-Петербург, 2003.
- [16] А.К. Нестеренко, А.А. Бездушный, Т.М. Сысоев, А.Н. Бездушный, В.А. Серебряков. Служба управления потоками работ по манипулированию ресурсами репозитория. // Электронный журнал, посвященный созданию и использованию электронных библиотек, том 6, выпуск 5. Москва: Институт развития информационного общества - 2003.
- [17] А.А. Бездушный, А.Н. Бездушный, А.К. Нестеренко, В.А. Серебряков, Т.М. Сысоев. Java и XML технологии новой версии ИСИР. // Сборник научных трудов X научно-практического семинара "Новые технологии в информационном обеспечении науки". Москва: 2003, с.182-205.

Distributed data integration based on the Semantic WEB and Workflow technologies

Nesterenko A.K. Sysoev T.M. Bezdushny A.A.
Bezdushny A.N. Serebriakov V.A.

This article describes one of the approaches to the distributed data integration problem solution. This technique is based on the open query concept, which uses workflow technologies in order to control an overall process of external system repositories querying and data aggregation. In this article some widespread data integration problems are considered and effective ways of their solution are provided.