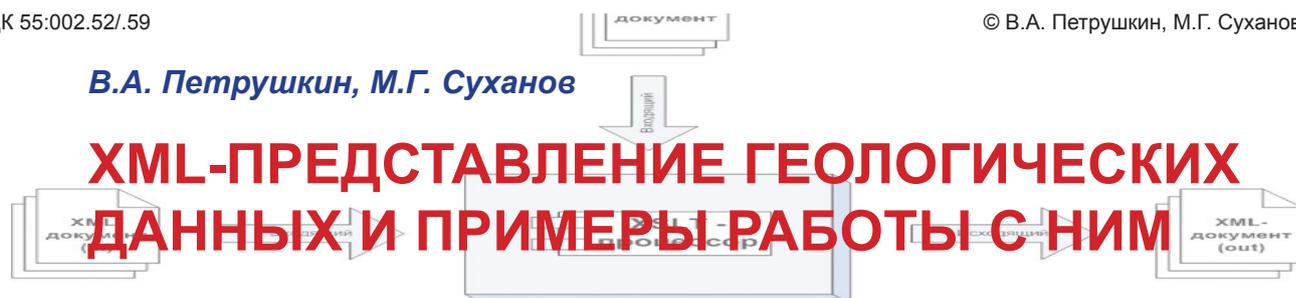


УДК 55:002.52/.59

© В.А. Петрушкин, М.Г. Суханов

В.А. Петрушкин, М.Г. Суханов



XML-ПРЕДСТАВЛЕНИЕ ГЕОЛОГИЧЕСКИХ ДАННЫХ И ПРИМЕРЫ РАБОТЫ С НИМ

Вопросы стандартизации в информационных технологиях и геоинформатике

Традиционно основной объем данных пользователи предпочитают хранить с использованием реляционных баз данных. Порядок организации хранения данных, как правило, диктуется удобством их занесения в базу, а также сильно привязан к используемым программным продуктам. Обращение к базе данных на каждом этапе работы с ними становится трудоемким процессом, так как информация о том, каким способом конечный пользователь будет их использовать, почти всегда неизвестна.

В тоже время современная геоинформационная система – сложная модульная система, состоящая из огромного количества библиотек, компонентов и программных продуктов от различных разработчиков, которые должны между собой обмениваться не только служебной информацией, но и данными. На этом фоне всплывает серьезная задача – определить, в каком виде необходимо хранить и представлять данные, чтобы обеспечить простоту и совместимость работы различных частей и компонентов систем.

Проблема обмена данными между информационными системами на разных этапах развития компьютерной техники и систем хранения решалась по-разному, но в конечном итоге мировое сообщество пришло к выводу, что наиболее конструктивное решение состоит в передаче данных от системы к системе в определенном формате, описанном общепризнанным стандартом. Только такой подход позволяет системам интегрироваться в общее информационное поле и поддерживать свою актуальность. Для решения этой задачи стали создаваться международные комитеты по стандартизации, сообщества разработчиков и появились первые стандарты, которые развивались и модернизировались.

Вдохновленные первыми успехами, разработчики стандартов из W3C (World Wide Web Consortium) предложили обобщить полученные достижения на широкий круг задач, создав более универсальный язык описания не только текстовой, но и любой другой информации различного назначения и степени сложности. Итогом этих работ явилось создание XML (eXtensible Markup Language –

расширяемый язык разметки). Это, видимо, самый прогрессивный и практически безальтернативный путь обеспечения совместимости и приведения данных к форме, делающей их пригодными для обмена между различными информационными системами и средствами разработки.

В настоящее время приложения и базы данных хранят информацию в самых разнообразных, порой несовместимых форматах. Обмен данными между несовместимыми системами – это одна из *наиболее неприятных и долгих процедур* для разработчиков.

С помощью XML можно осуществить обмен данными даже между несовместимыми системами. Преобразование данных в XML-формат позволяет упростить эту задачу, сделать данные доступными для самых различных типов программ.

Логика проста – все передаваемые данные находятся в текстовом виде, который может быть прочитан любым текстовым редактором на любой платформе. Группировка данных производится с помощью служебных слов – XML-элементов. На синтаксис самих XML-элементов накладывается лишь ряд незначительных ограничений. XML-элемент – составная часть XML-документа, представляющая собой некоторую законченную смысловую единицу, которая может содержать один или несколько вложенных элементов и/или атрибутов. Атрибут представляет собой составную часть элемента, задающую его параметры. Имена элементов и атрибутов являются регистрозависимыми.

Фактически любая иерархическая модель может быть описана в удобной для поставщика данных форме с помощью им же созданных XML-документов и XML-схем (XML Schema), описывающих эту модель. Любой получатель данных может однозначно интерпретировать их с помощью XML-схемы, которую он получает вместе с XML-документом, либо по ссылке на ресурс, где эта схема хранится.

XML Schema – один из языков описания структуры XML-документа. Спецификация XML Schema является рекомендацией W3C и создана, чтобы играть центральную роль в последующей разработке XML, особенно в веб-сервисах, в которых она является одним из фундаментальных принципов.

XML Schema делает следующее:

- описывает названия элементов и атрибутов (словарь);
- описывает взаимосвязь между элементами и атрибутами, а также их структуру (модель содержания);
- описывает типы данных.

При помощи схем можно описывать практически все требования к XML-документу. То есть, схема – это универсальный способ описания грамматики данных, который может применяться не только для верификации XML-документов, но и описания баз данных и т.д. Таким образом, область применения схем на данный момент очень широкая. Каждый элемент в этой модели ассоциируется с определенным типом данных, позволяя строить в памяти объект, соответствующий структуре XML-документа. Объектно-ориентированным языкам программирования гораздо легче иметь дело с таким объектом, чем с текстовым файлом.

Другим удобством XML-схемы является то, что один словарь может ссылаться на другой. Таким образом, разработчик может использовать уже существующие словари и ему легче устанавливать и распространять стандарты XML-структуры для определенных задач.

В связи с резко возросшей популярностью и объемами передаваемой таким способом информации, а также ее разнородностью, для облегчения обработки полученных данных перед мировым сообществом встала задача стандартизации структуры передаваемой информации. Начали разрабатываться и внедряться различные внутриведомственные и межотраслевые стандарты, которые наиболее полно описывали бы различные предметные области для ускорения получения, обработки и интерпретации и передачи такой информации внутри специализированных сообществ.

В области ИТ (Information Technology) все стандарты можно формально разделить, в зависимости от порядка разработки и процедуры принятия, на три категории:

- стандарты, разработанные и утвержденные официально;
- общеупотребительные стандарты при отсутствии официального статуса;
- стандарты, разрабатываемые консорциумами.

Считается, что последние отличаются относительно большой скоростью разработки и эффективной процедурой принятия, являя собой удачную альтернативу двум предыдущим типам. Российское законодательство по стандартизации существование таких стандартов допускает.

Что касается официальных стандартов, то вопросами их разработки в области пространственной

информации занимается множество организаций в разных странах. В России это Государственный комитет РФ по стандартизации и метрологии (Росстандарт, www.gost.ru), Федеральная служба геодезии и картографии России (Роскартография) перешедшая в структуру Росреестра, Государственный научно-внедренческий центр геоинформационных систем и технологий (Госгисцентр), а также Центральный научно-исследовательский институт геодезии, аэро съемки и картографии им. Ф.Н. Красовского (ЦНИИ-ГАиК) и 29-й Научно-исследовательский институт МО РФ. В структуре Росстандарта проблемами геоинформатики занимается Технический комитет по стандартизации ТК22 «Информационные технологии». Часть стандартов разработана Техническим комитетом по стандартизации ТК-394 «Географическая информация/геоматика».

В мировых масштабах основное влияние в области разработки официальных стандартов пространственной информации имеют ISO (International Standard Organization, www.iso.org), американские FGDC (Federal Geographic Data Committee, www.fgdc.gov), USGS (US Geological Service, www.usgs.gov), ANSI (American National Standards Institute, www.ansi.org) и NIMA (National Imagery and Mapping Agency, www.nima.mil), International Cartographic Association (ICA, www.geovista.psu.edu/sites/icavis) и некоторые другие.

Среди этого списка особо выделяется ISO, в состав которой входит технический комитет 211 (ISO/TC 211, www.isotc211.org), занимающийся вопросами геоматики и стандартизации географической информации. В России с ISO/TC 211 работает ТК-394.

OGC (Open Geospatial Consortium) является международным консорциумом, в который входят более 400 компаний, правительственных агентств и университетов, участвующих в процессе согласования проектов разработки стандартов, протоколов и требований к обработке геоинформационных данных. Эти стандарты, протоколы и требования поддерживают взаимодействие различных систем во всемирной сети, систем мобильной связи и локализованного сервиса (LBS), способствуют распространению ИТ и помогают разработчикам обеспечивать пользователям доступ к комплексной геопространственной информации и услугам.

OGC разрабатывает третью категорию стандартов. Необходимость в таком типе стандартизации определяется проблемами, связанными с качеством официальных стандартов, а также длительным сроком их принятия.

В 1999 году ISO опубликовала соглашение между ISO/TC 211 и OGC, которое формализовало

вопросы сотрудничества между двумя организациями. С тех пор взаимодействие между ISO и OGC продолжает расширяться и углубляться.

Обзор некоторых XML-стандартов в геологической отрасли

На данный момент создано довольно большое количество стандартов, основанных на XML-представлении данных, которые отражают те или иные потребности геологической отрасли, в том числе и специфические. Некоторые из них имеют узкое применение. Другие, наоборот, получили широкое распространение. Разработкой и утверждением спецификаций, как правило, занимаются группы компаний, ассоциации, консорциумы, а также различные комитеты по стандартизации и унификации.

Среди наиболее заметных и распространенных форматов можно отметить следующие.

KML (Keyhole Markup Language – язык разметки Keyhole) – язык разметки на основе XML, разработанный компанией «Keyhole» до ее приобретения компанией «Google». KML – это формат представления и хранения данных, который используется для отображения трехмерных геопространственных данных в средствах просмотра этого формата, например Google Планета Земля, Карты Google и Карты Google для мобильных устройств. KML создан на основе формата XML и использует основанную на тегах структуру с вложенными элементами и атрибутами. KML представляет собой открытый стандарт кодирования. Он поддерживается консорциумом OGC.

GML (Geography Markup Language – язык географической разметки), разрабатываемый OGC. GML основан на XML-кодировании пространственной и атрибутивной информации. Этот формат позволяет пользователям организовать прозрачную и стандартную среду для публикации и обмена геоинформацией, главным образом в сети Интернет.

Авторитетный международный институт в сфере стандартизации нефтегазовых технологий, хранения и управления геолого-геофизическими данными и информационного обмена Petrotechnical Open Standards Consortium (POSC) выступил с рядом инициатив по принятию XML-стандартов в области нефтегазовой отрасли:

WITSML (Wellsite Information Transfer Standard Markup Language) – стандарт языка разметки для передачи буровой информации [1].

PRODMML (Production Markup Language) – стандарт языка разметки для обмена данными, используемыми для мониторинга, управления и оптимизации добычи углеводородов [2].

RESQML (Reservoir Characterization Markup Language) – стандарт языка разметки для обмена данными для определения характеристик коллекторов углеводородов, земли и модели резервуара [3].

Стандарты не были официально приняты, однако получили широкое распространение в отрасли.

В Великобритании в марте 2007 года было проведено стартовое совещание проекта OneGeology [4]. В общей сложности 81 геолог из 43 стран мира собрались, чтобы рассмотреть возможность создания динамической бесшовной цифровой геологической карты мира. Проект международный и осуществляется под эгидой нескольких международных организаций, таких как Международный союз геологических наук (МСГН), Международный руководящий комитет по глобальному картографированию (МПКГК), ЮНЕСКО. Целевой масштаб геологической карты 1:1 000 000.

В рамках этого проекта был разработан стандарт языка GeoSciML, представляющий собой основу для независимого представления цифровых геологических данных и связанных с ними пространственных данных. В настоящее время пределы проработки формата ограничены информацией, наиболее часто отображаемой на геологической карте, включая скважины и данные полевых наблюдений.

OneGeology руководствовался следующим набором принципов, требующих от геологов и геофизиков во всем мире:

- доступность к лучшим геологическим данным и картам в любом цифровом формате, имеющимся в каждой стране;
- участие в последовательной разработке стандартов для доступа к данным – схем взаимодействия. Передача необходимых разработок тем, кто в них нуждается, принимая во внимание тот факт, что разные страны имеют различные способности и возможности участия в проекте;
- стимулирование быстрого увеличения совместимости, которое достигается за счет разработки и использования языка разметки GeoSciML.

EarthResourceML – формат представления информации о минеральных ресурсах, технологических особенностях их исследования и физико-химических свойствах. Основой стал формат MineralOccurrence, разработанный для описания характеристик и свойств твердых полезных ископаемых. В последствии MineralOccurrence стал вместе с рядом других форматов частью более общего описания EarthResourceML. Он также является XML-форматом.

XMML (eXploration and Mining Markup Language) – XML-спецификация для передачи через Интернет данных геологоразведки и данных о добыче полезных ископаемых горнодобывающей промышленностью. XMML разрабатывается как приложение Geography Markup Language (GML), который описывает геометрию для использования в конкретных областях. Взаимодействие с GML позволяет XMML обеспечить совместимость передаваемых данных от серверов к клиентам с помощью веб-сервисов.

Некоторые технологии работы с XML-документом.

Обоснование выбора XSLT-процессора SAXON и его адаптация для работы с базами данных

Разметка типичного XML-документа описывает его структуру, но старается не затрагивать его представление, то есть говорит о том, как он организован, а не как он выглядит. Хотя XML-документы являются текстовыми и человек при желании может их читать в исходном виде, обычно XML-документ преобразуют в другой формат перед тем, как представить публике. Ключевая идея языков разметки вообще и XML в частности состоит в том, что входной формат не обязан совпадать с выходным форматом.

Входные языки разметки разработаны для удобства автора. Выходные форматы создаются для удобства пользователя.

Фактически процесс работы с XML выглядит следующим образом:

1. Создается XML-документ, содержание которого описывается по схеме пользователя в соответствии с XML-схемой, либо с применением одного из стандартных языков, разработанных для работы с XML и поддерживающих его структуру.
2. Запускается XML-документ в режиме просмотра (если его назначение – представление данных), либо в режиме исполнения (если его назначение – преобразование данных). При запуске XML-документа в режиме исполнения результатом работы на выходе является либо новый XML-документ, либо поток данных, отражающий структуру выходного XML-документа.

Наиболее распространенные стандарты для работы с XML-структурой документа следующие:

XPath – декларативный язык, который используется для обозначения частей XML-документов. Выражения XPath используются для определения местоположения набора узлов в

определенном XML-документе. Многие технологии XML, такие как XSLT и XQuery, широко используют XPath.

XLink – декларативный язык, позволяющий вставлять в документы XML-элементы, создающие связи между XML-документами и описывающие их внутри самих документов.

XQuery – декларативный язык для выполнения запросов к структурированным или полуструктурированным XML-данным.

XSL (Extensible Stylesheet Language – расширяемый язык таблиц стилей) делится на две части: XSL-трансформации (XSL Transformations, XSLT) и форматирующие объекты XSL (XSL Formatting Objects, XSL-FO). XSL-таблица стилей – более мощный и гибкий инструмент для отображения XML-документов, чем CSS-таблица. Используя XSL-таблицы стилей, можно не только задать формат для каждого элемента XML-документа, но и обеспечить средства контроля над выводимыми данными. XSL позволяет выбрать те данные из XML-документа, которые вы хотите отобразить, в любом порядке представить и свободно модифицировать, а также добавить новую информацию.

XSLT – это XML-приложение, определяющее правила, в соответствии с которыми один XML-документ преобразуется в другой. XSL-документ, то есть XSL-таблица стилей, содержит шаблоны, которые XSLT-процессор сравнивает с элементами входного XML-документа. Когда соответствующий шаблон найден, процессор записывает содержимое шаблона в выходное дерево. По завершении этого этапа выходное дерево записывается либо в XML-документ, либо в другой формат, такой как обычный текст или HTML. Конечно, такая схема требует средств для преобразования входного формата в выходной. Большинство XML-документов подвергаются трансформации перед представлением потребителю.

XSLT-процессор – это программа, которая читает XSL-таблицу стилей, читает входной XML-документ и преобразует входной документ в выходной в соответствии с инструкциями, описанными в таблице стилей.

XSLT-процессор может быть встроенным в веб-браузер, как MSXML в Internet Explorer. Он может быть встроен в веб-сервер или в сервер приложений, как Cocoon в проекте Apache XML (xml.apache.org/cocoon). Либо он может быть отдельной программой, запускаемой из командной строки, как Saxon [5] Майкла Кэя (Michael Kay) или Xalan из проекта Apache XML (xml.apache.org/xalan).

Конкретные детали установки, настройки и запуска различны у разных XSLT-процессоров.

Они описываются в документации, поставляемой с XSLT-процессором.

При работе с XML-документом наиболее перспективным представляется XSLT-процессор Saxon версии 8.x. Этот выбор обусловлен тем, что на данный момент Saxon является чуть ли не единственным процессором, который поддерживает спецификацию XSLT 2.0 [6] и XPath 2.0 [7] почти в полном объеме и даже рабочую версию спецификации XSLT 1.1. Общий принцип реализации технологии и функционирования XSLT приведен на рис. 1.

Преимущества использования XSLT версии 2.0 по сравнению с предыдущей версией 1.0 огромны. В первую очередь новая спецификация основана на XPath 2.0, что позволяет использовать весь набор дополнительных встроенных функций этой спецификации по работе с XML-элементами. Реализована возможность вывода нескольких выходных документов, а также реализован вывод их содержимого не только в XML-, Text- и HTML-виде, но и в виде правильно оформленного XHTML. В XSLT 1.0 отсутствовала встроенная поддержка группировки. В XSLT 2.0 она реализована с помощью простого выражения:

```
<xsl:for-each-group select=".....">.....</xsl:for-each-group>
```

Процессор Saxon позволяет обращаться к функциям-расширениям, написанным на Java, следуя интерфейсу, который был определен в проекте стандарта XSLT 1.1. Еще одним серьезным преимуще-

ством Saxon является наличие как Windows-реализации, так и версии под Linux. Saxon работает в любой среде, где есть JVM (Java Virtual Machine). Кроме того, имеются реализации процессора Saxon для платформы .NET.

На момент написания статьи был выпущен процессор Saxon версии 9.x, ориентированный на поддержку спецификации XSLT 3.0, однако его бесплатная версия уже является сильно урезанной и поэтому актуальность Saxon 8.x продержится еще долго.

В качестве примера рассмотрим автоматизацию получения данных напрямую из базы данных через расширение, реализованное внутри XSLT-процессора Saxon.

В настоящее время Saxon в качестве языка расширения поддерживает только Java, поэтому предпочтительный способ объявления расширения в документе XSLT-кода – объявить на верхнем уровне пространство имен в виде java: и далее полностью квалифицированное имя класса, как Java метод в пространстве имен. Например,

```
xmlns:sql="java:/net.sf.saxon.sql SQLElementFactory".
```

С помощью Java-расширения функционала Saxon удастся напрямую осуществлять подключение к базам данных и формировать XML, наполняя его извлеченными данными, минуя промежуточные операции по работе с базами данных, такие как сохранение извлеченных данных и их преобразование в различные форматы.

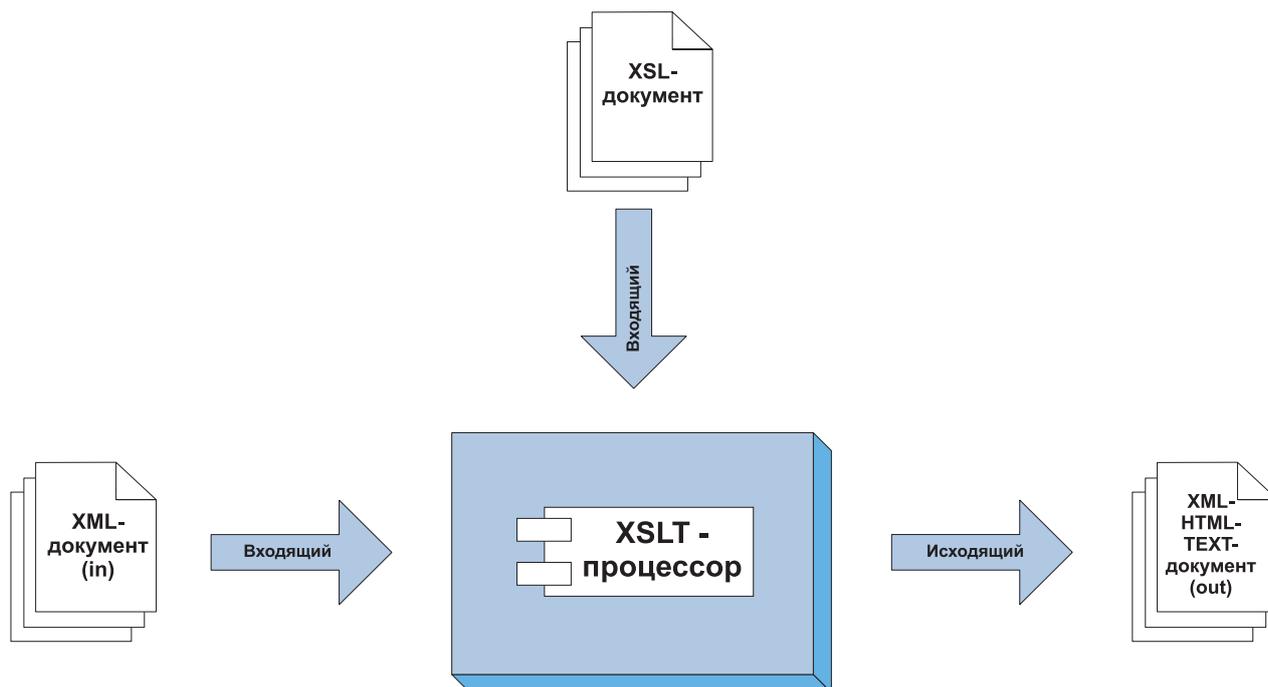


Рис. 1. Общий принцип работы XSLT-преобразования применительно к XML

ВНИИгеосистем в рамках работ по КИПС ГБЦГИ [8] был реализован веб-сервис, который представляет данные из базы данных MS SQL Server в XML-формате со сложной структурой (в соответствии с ИСО 19139 представления метаданных). Формирование этой структуры осуществляется запуском в процессоре Saxon самого XSL-документа с XSLT-кодом и настройкой соединения с базой данных через ODBC (Open Database Connectivity) и/или JDBC (Java Database Connectivity).

Описание реализованного Java-расширения для установления прямого соединения с удаленной базой данных MS SQL Server с помощью параметра «database» через JDBC или же через ODBC выглядит так:

```
<xsl:param name="database"
select="jdbc:jtds:sqlserver://192.168.203.115/MetaDBsql;
instance=STORM2005"/>
```

Соединение с локальной базой данных, например MS Access:

```
<xsl:param name="database" select="jdbc:odbc:
Meta_izuch_data" />
```

Отдельно описывается параметр «driver»: Для MS SQL Server:

```
<xsl:param name="driver" select="net.sourceforge.
jtds.jdbc.Driver"/>
```

Для MS Access:

```
<xsl:param name="driver" select="sun.jdbc.odbc.
JdbcOdbcDriver"/>
```

Полученное решение обладает рядом преимуществ, так как дает возможность уйти от необходимости вручную управлять соединением с базой данных и избежать ручного формирования входящего XML-документа и направления его на вход XSLT-процессора. Это экономит большое количество времени и приводит к сокращению количества ошибок на этапе подготовки. Далее определяется переменная, в которой будут храниться все параметры запроса и указывается пространство имен языка функции-расширения. Работа с этой переменной уже не зависит от происхождения данных. Само обращение к этой переменной приводит к автоматическому установлению соединения с базой.

```
<xsl:variable name="connection" as="java:java.lang.Object"
xmlns:java="http://saxon.sf.net/java-type">
<sql:connect driver="{driver}" database="{database}"
user="{user}" password="{password}"/>
</xsl:variable>
```

После этого дальнейшая работа кода никак не связана с платформой, на которой используется XSLT-процессор. Данные, с которыми предстоит работа, XSLT-программа воспринимает как обычную переменную. Схема работы XSLT-преобразования продемонстрирована на рис. 2.

Такое решение успешно применяется в рамках работ по КИПС ГБЦГИ для задачи унификации представляемых данных о полезных ископаемых. Была создана вспомогательная база данных, разработанная на основе «Общероссийского классификатора полезных ископаемых и подземных вод» (ОКПИИПВ)

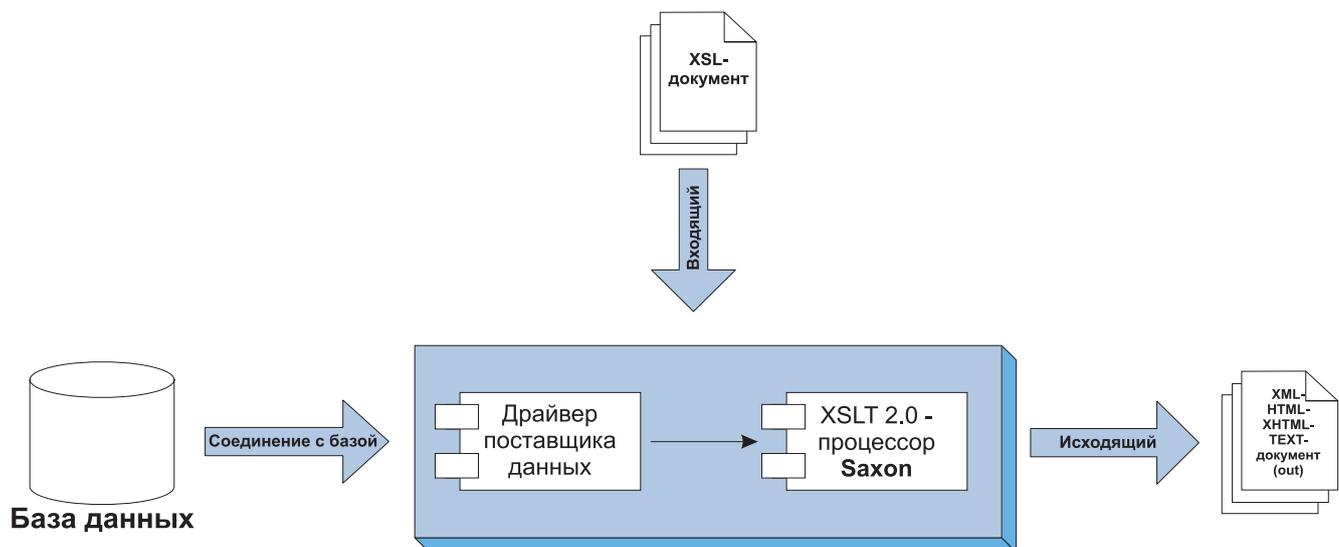


Рис. 2. Уточненная схема работы XSLT-преобразования применительно к специфике выполняемой работы

ОК 032-2002 [9], реализованная в среде MS Access. С ее помощью удалось частично структурировать сведения о наличии полезных ископаемых, которые носили в исходных материалах описательный характер в виде обычного текста, и это не позволяло производить даже частичный поиск по полезному ископаемому. Соединение с базой данных осуществлялось при помощи описания параметров соединения и обращения к базе данных через ODBC непосредственно из кода программы XSL-документа. В результате на выходе получали структурированный XML-документ, содержащий сопоставление полезного ископаемого и его описания в классификаторах, дополненный метаданными.

Включение в XSLT-преобразование дополнительного шаблона и программного кода, реализующего разбор текста описания полезных ископаемых, отбор и сопоставление с сгруппированными данными, позволило частично (примерно на 70%) классифицировать описываемые в исходных материалах полезные ископаемые. После выполнения XSLT-преобразования в выходной XML-документ добавляется секция с указанием полезного ископаемого и ссылкой на классификатор. При необходимости результат по полезным ископаемым может быть выведен как в виде ори-

гинального текста (рис. 3), так и дополнительно в XML-виде, с секцией, содержащей классификацию по (ОКПИиПВ) ОК 032-2002:

```
<geologicalInfo>
  <GX_MineralIdentification>
    <mineral>
      <GX_MineralCode codeList="rfgf.ru:isnedra:gkm"
        codeListValue="341">вольфрам</GX_MineralCode>
    </mineral>
    <GX_MineralCode codeList="okpipv"
      codeListValue="132081">Руда вольфрамовая и песок на
        вольфрам</GX_MineralCode>
    </mineral>
  </GX_MineralIdentification>
</geologicalInfo>
```

Для создания запросов к базе данных ORACLE, в рамках сотрудничества ВНИИгеосистем со ВСЕ-ГЕИ (Всероссийский научно-исследовательский геологический институт им. А.П. Карпинского), опробовано еще одно решение. Через пользовательскую функцию в XSL-документе была реализована возможность использования PHP-функций внутри XSLT-кода.

Управляющие SQL-команды представляют собой просто текстовую строку определенного

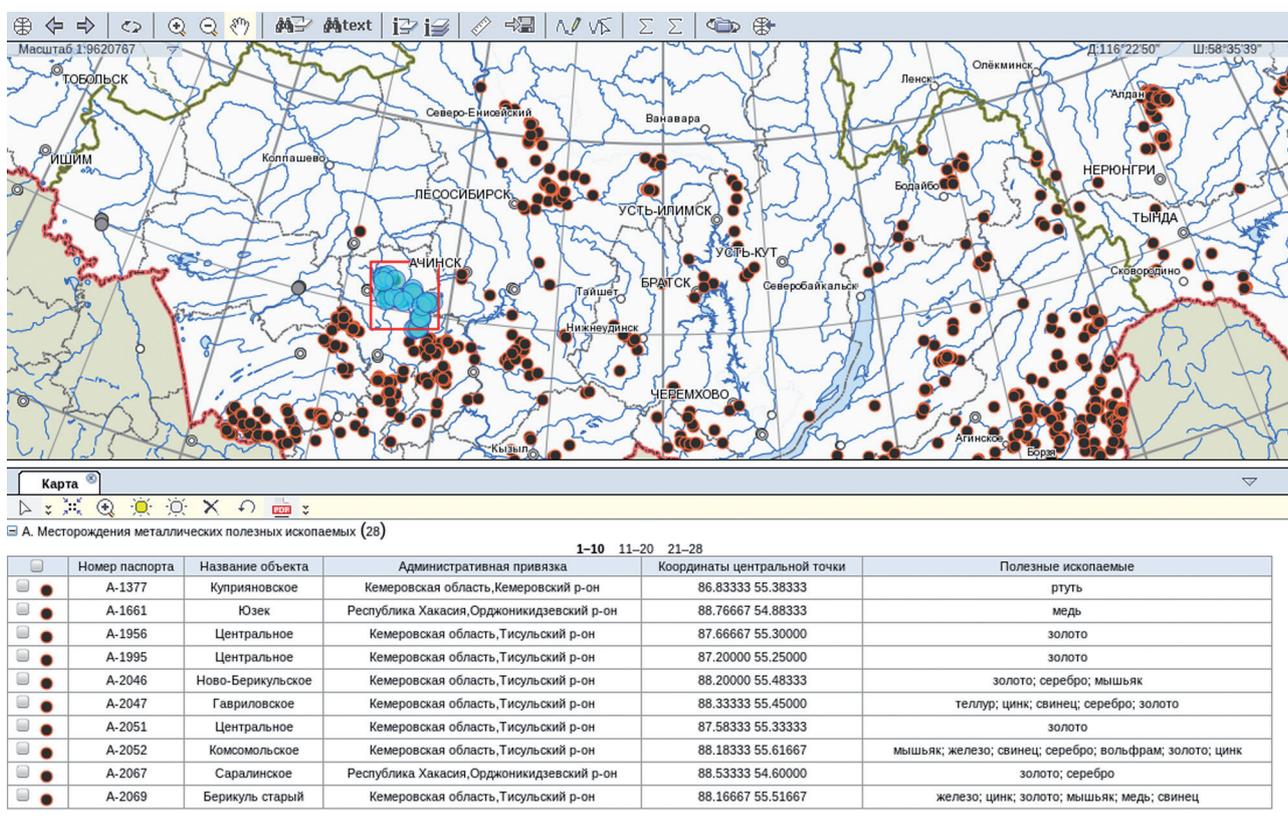


Рис. 3. Результат работы коннектора для государственного кадастра месторождений

синтаксиса. Для их выполнения была написана XSLT-функция, передающая в функцию PHP-сценария параметр в виде текста SQL-запроса. Внутри PHP-сценария описано установленное соединение с базой данных ORACLE и возврат результата выполнения SQL-запроса в виде XML-выражения. Полученные данные заносились в переменную, и дальнейшая работа кода уже не зависела от того, каким способом они были получены.

В случае использования PHP-функций XSLT-программа обеспечивает независимую работу с любым XSLT-процессором и на любой платформе, необходимо лишь один раз настроить среду исполнения. Это решение более универсальное, однако оно требует определенных затрат на программирование PHP-функций, корректно обеспечивающих установление связи с базой данных, выполнение SQL-запроса и возврат результата, а кроме того, описание корректной обработки возникающих исключений при работе с базой данных. Схема взаимодействия с базой данных изображена на рис. 4.

Синтаксически XSLT-реализация обращения к PHP-функции и занесение в переменную данных запроса выглядит, например, так:

```

xmlns:php="http://php.net/xsl"
.....
<xsl:variable name="R_COA_NAME"
select='php:function("sql_query",concat("select COA_NAME_FR from
NADM_USER.COA where COA_ID=", $MAP_UNIT//UNIT_RANK))'/>
    
```

В целом комбинация вышеизложенных технологий обеспечивает универсальное кроссплатформенное решение целого класса задач, связанных с представлением данных, хранящихся в оптимизированном под SQL-запросы виде, как иерархической структуры любой степени вложенности, минуя целый ряд вспомогательных операций. Основные трудозатраты заключаются в настройке параметров среды исполнения.

Заключение

Описанный в статье подход расширяет возможности работы с базами данных, т.к. позволяет представить хранящуюся в них информацию в виде иерархической модели в XML-формате с любой степенью вложенности. Использование XML и инструментария для работы с ним позволяет более эффективно взаимодействовать с полуструктурированными (не имеющими жестко фиксированной структуры

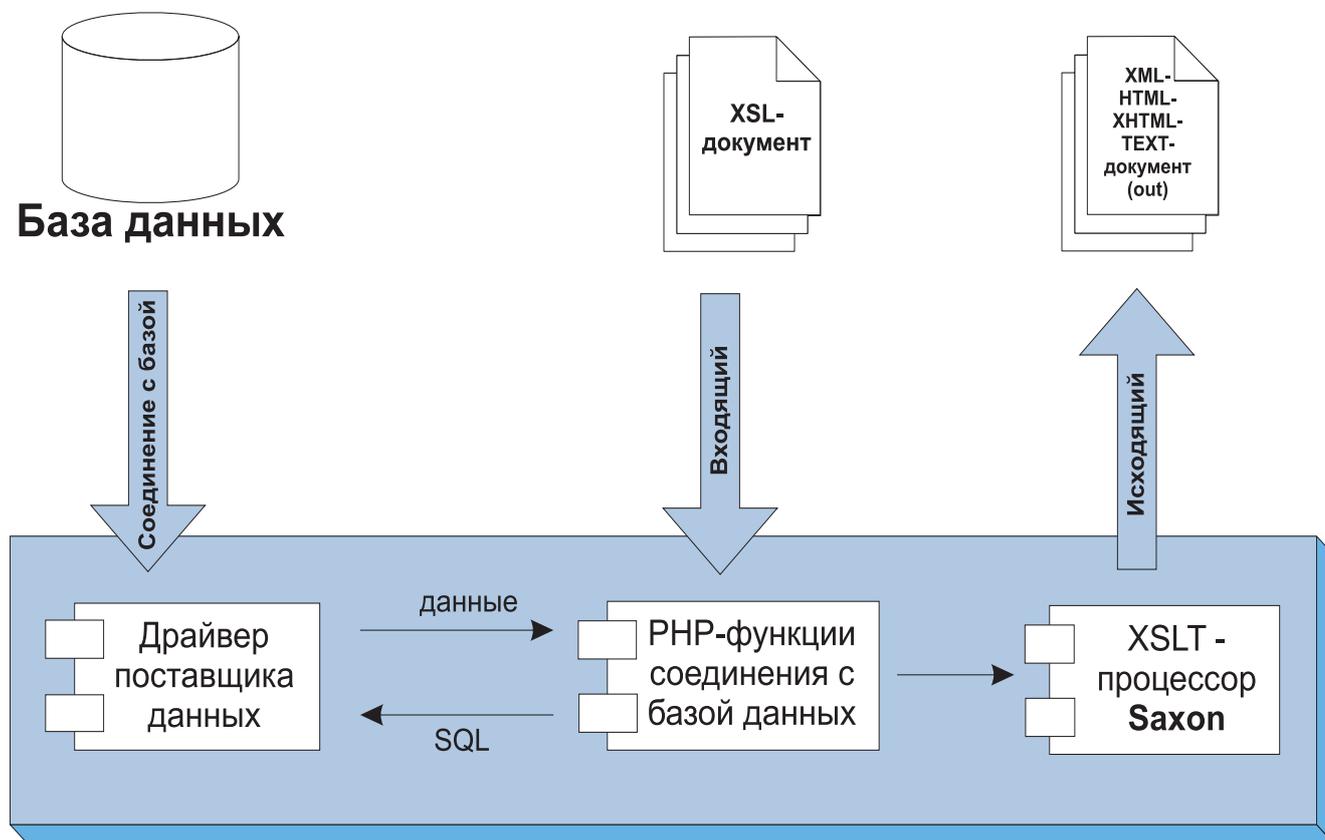


Рис. 4. Схема взаимодействия с базой данных через соединение, реализованное в PHP-функциях

представления) данными и эффективно использовать стандартные операции поиска, группировки, отбора и сортировки данных с большей эффективностью, чем с использованием аналогичных SQL-операций. Автоматизация взаимодействия с базой позволяет максимально исключить человеческий фактор практически до момента получения конечного результата. Один раз, произведя разработку и отладку содержимого исполняемого текста XSL-документа, можно, меняя источники данных, описанные в соединении с базой данных, многократно получать конечный результат в заранее определенном виде. Стандартные же средства, предусмотренные в самих базах данных по формированию XML-структуры, как правило, не могут обеспечить множественное вложение и группировку.

Предложенные решения являются достаточно универсальными и применимыми к целому классу задач, по обработке большого количества данных, извлекаемых непосредственно из баз данных. Несмотря на разницу в структуре хранения данных в реляционных базах данных и XML-данных, полученные результаты вполне удовлетворительны, что подтверждает обоснованность применения данного подхода.

XML-представление данных имеет еще одно огромное преимущество. Этот формат открытый и никак не привязан к платформе. То есть любой текстовый редактор на любой платформе позволяет получать данные XML. А любой разборщик и любой XSLT-процессор может обработать данные в XML-виде.

Поэтому способ представления конечных данных пользователю в XML-формате представляется наиболее удобным и перспективным, обеспечивающим гибкость работы с ними. Поэтому работы в этом направлении ведутся весьма успешно во многих странах мира, что расширяет возможности интеграции в международные разработки.

Ужесточение требований к способам обработки, хранения, передачи и представления данных – мера, также способствующая упрощению их взаимной интеграции между различными информационными системами. Разработка и внедрение единой общей инструкции, регламентирующей этот процесс в отрасли, а в идеале создание единого внутриотраслевого обменного формата, который бы отражал нужды всех заинтересованных сторон –

перспективный и прогрессивный подход, хорошо зарекомендовавший себя во всем мире практически во всех отраслях.

Большое количество спецификаций и стандартов отображения в XML-формате геологических данных, а также предусмотренная возможность их взаимного включения друг в друга обеспечивают большую гибкость в решении самых разноплановых задач, охватывающих практически все аспекты геологической отрасли.

Ключевые слова: XML, XSLT, Saxon, ODBC, JDBC, PHP, КИПС ГБЦГИ, ВСЕГЕИ.

ЛИТЕРАТУРА

1. WITSML Standards // Energistics : The Energy Standards Resource Center [Electronic resource]. – URL: <http://www.energistics.org/drilling-completions-interventions/witsml-standards> (date of access 12.07.2012).
2. PRODML Standards // Ibidem. – URL: <http://www.energistics.org/production/prodml-standards> (date of access 16.07.2012).
3. RESQML Standards// Ibidem. – URL: <http://www.energistics.org/reservoir/resqml-standards> (date of access 16.07.2012).
4. OneGeology: Making Geological Map Data for the Earth Accessible [Electronic resource]. – URL: <http://www.onegeology.org> (date of access 02.08.2012).
5. Saxonica: XSLT and XQUERY processing [Electronic resource]. – URL: <http://www.saxonica.com> (date of access 07.09.2012).
6. XSLT 2.0 and XQuery 1.0 Serialization (Second Edition) [Electronic resource]. – URL: <http://www.w3.org/TR/2010/REC-xslt-xquery-serialization-20101214> (date of access 02.10.2012).
7. XML Path Language (XPath) 2.0 (Second Edition) [Electronic resource]. – URL: <http://www.w3.org/TR/xpath20> (date of access 02.10.2012).
8. Общероссийский классификатор полезных ископаемых и подземных вод : ОК 032-2002. – М., 2012. – URL: <http://stat.kostroma.ru/method/Lists/Class/Attachments/17/ОКПИПВ.pdf> (date of access 04.10.2012).
9. Картографическая информационно-поисковая система Государственного банка цифровой геологической информации (КИПС ГБЦГИ). URL: <http://portal.gbcbgi.geosys.ru> (date of access 06.10.2012).