

## ПРИМЕНЕНИЕ XML-ПРЕДСТАВЛЕНИЙ ДЛЯ ИНТЕГРАЦИИ ДАННЫХ В ПРОМЫСЛОВОЙ ГЕОФИЗИКЕ

*В.Г. Тульчинский, П.Г. Тульчинский, А.К. Ющенко, Р.А. Ющенко*

Институт кибернетики имени В.М. Глушкова НАН Украины,  
03687, Киев, проспект Академика Глушкова, 40, тел. 526 3603,  
pgt@ukr.net

Описана проблема интеграции данных геофизических и геологических исследований. Проанализированы обменные геофизические форматы, основанные на XML. Рассмотрена реализация графовых запросов для формирования XML-представлений и применение XQuery для их объединения.

The problem of the geophysical and geological data integration is described. The exchange geophysical formats based on XML are analyzed. The implementation of graph queries to form XML-representations and application of XQuery for their integration are examined.

### Введение

Интерпретация данных геологических исследований также предполагает интенсивный обмен информацией между программными системами различных производителей. Для успешного использования форматы разных систем должны поддерживаться каждой из этих систем, либо сводиться к формату, который всеми поддерживается. Учитывая постоянное расширение номенклатуры геофизических методов, внедрением новой аппаратуры и технологий, в геофизике постоянно актуальна задача обмена данными. В данный момент наиболее популярным способом обмена геологическими данными являются отформатированные текстовые файлы. В отдельных случаях применяются средства импорта /экспорта, позволяющие синхронизировать реляционные схемы данных. Но во всех случаях для успешного обмена необходимо соотнести схему данных источника со схемой данных получателя. И так – для каждой пары взаимодействующих систем.

Для решения проблемы связи между системами, работающими с геологическими данными, а так же для организации информационного хранилища удобно использовать XML. Это позволяет создать мобильный независимый от операционной системы, СУБД и приложений формат данных, который удобно проверять и интерпретировать стандартными средствами.

Промысловая геофизика, связанная с разведкой и эксплуатацией месторождений полезных ископаемых, традиционно сильно зависит от прогресса вычислительной техники и программного обеспечения, оказывая не меньшее влияние на информационно-коммуникационные технологии со своей стороны. Цена получения информации о локальной структуре земных недр очень высока, стоимость ошибки при бурении одной скважины – несколько миллионов долларов, поэтому постоянно совершенствуется аппаратная база, методика измерений, обработки и интерпретации данных, программное обеспечение. В результате сервисным компаниям, предоставляющим геофизическую информацию, и ее потребителям, добывающим предприятиям, не хватает одной интегрированной системы. Они одновременно используют разные программные продукты.

И сервисные компании и тем более добывающие предприятия давно стремились достичь интероперабельности данных на всех четырех уровнях [1]: системном, синтаксическом, структурном и семантическом (рис. 1). Системный уровень, включающий сетевые протоколы, Web-сервисы и операционные системы почти не затрагивается отраслевой спецификой. Синтаксический уровень представлен стандартными форматами данных и языками спецификаций. Этот уровень в промышленной геофизике довольно долго был представлен общепризнанными в отрасли стандартами «de facto», такими как LAS, SEG-Y, LIS, SPS. Но синтаксическая совместимость еще не гарантирует автоматического разбора описанной этим синтаксисом информации. Начиная с 1980-х удачные инициативы в достижении интероперабельности, такие, как всемирно признанная система стратиграфической классификации отложений [2] или обменный формат данных бурения скважин WITS [3], фокусировались на определении как можно более широкого круга уникально и единообразно интерпретируемых понятий (семантический уровень).



Рис. 1. Уровни интероперабельности

На структурному рівні формати, подібні WITS, залишалися неудовлетворительними. Вони підтримували дуже ограничену номенклатуру об'єктів і властивостей.

Перші зусилля по розвитку виразительних можливостей галузевих форматів, такі як версія 3.0 формату LAS [4], не змогли сдвинуть старі прості формати із-за переусложненности і відсутності стандартних програмних інтерфейсів. Тільки з поширенням XML у нового покоління обмінних форматів з'явилися можливості представляти складні структури геофізических даних розвідки і експлуатації родовищ корисних копалин. Наприклад, WITSML [5] був первісно розроблений як

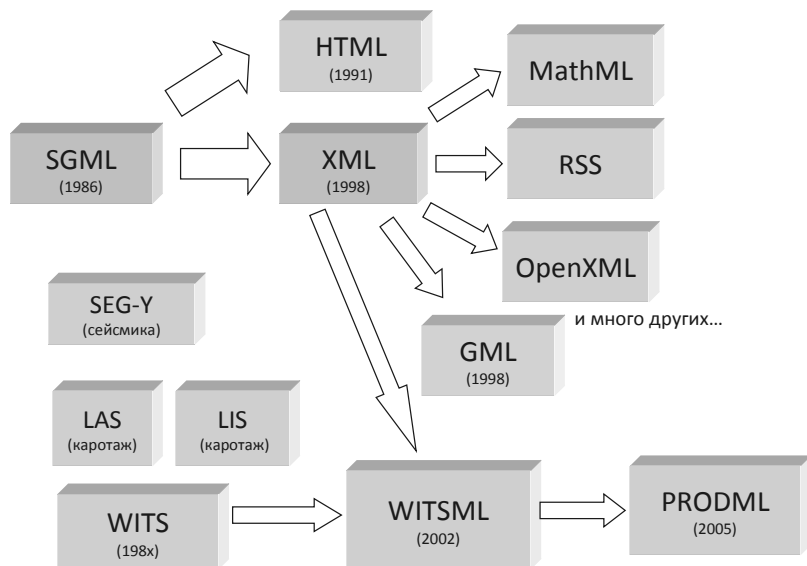


Рис. 2. Обменные форматы геофизических данных

новая, рассчитанная на интернет-технологии версия широко используемого формата WITS. Но эта разработка, основанная на стандартах W3C, применении XML и сервис-ориентированном программном интерфейсе SOAP быстро преодолела ограничения, свойственные WITS. PRODML [6] отпочковался от WITSML как формат для поддержки оптимизации добычи нефти и газа. Открытый консорциум геологического пространства (Open Geospatial Consortium) разработал основанные на XML обменные форматы XMML [7] и GeoSciML [8]. Оба эти формата опираются на ISO/DIS 19136 GML [9]. В течение последних лет возникло множество других современных форматов на основе XML [10]. Основные обменные форматы показаны на рис. 2.

Рассмотрим подробнее роль XML в повышении итероперабельности приложений промышленной геофизики. В качестве примера проанализируем два очень близких по составу и назначению формата: WITS и его новую, основанную на XML, версию WITSML.

### Основанные на XML стандарты обменных форматов промышленной геофизики

WITS – это Стандарт передачи информации со скважины (Wellsite Information Transfer Standard). Он используется для автоматизации удаленного обновления в центральной БД компании технологических данных процесса бурения, информации о разбуриваемых породах и данные геофизических исследований непосредственно в процессе проходки и после ее окончания вплоть до цементирования колонны. WITS разбит на несколько таких уровней:

- уровень 0: ASCII-формат; включает фиксированные предопределенные форматы записей;
- уровень 1: Бинарный формат; аналогичен ASCII-формату, введен для компактности;
- уровень 2: Ограниченный двухсторонний диалог через записи-комментарии;
- уровень 2b: Буферизация, чтобы записи не терялись при разрыве коммуникационного канала;
- уровень 4: Расширяемый формат записей RP66.

Несмотря на долгую историю и широкое распространение WITS обладает некоторыми ограничениями, свойственными большинству традиционных EDI-форматов. Эти ограничения можно объединить в две группы.

1. Недостаточная поддержка новых решений в инженерии скважин:
  - устаревшие форматы записей (особенно относящихся к каротажу во время бурения);
  - ограниченное число разделов под бурильные колонны;
  - отсутствие гибкости в использовании различных единиц измерения;
  - почти полное отсутствие поддержки статической (справочной) информации о скважине;
  - ограниченная возможность двунаправленного диалога для передачи управленческих решений на скважину.
2. Недостаточная поддержка современных Web-ориентированных программных решений:
  - бинарный формат (уровень 1) остается платформо-зависимым;
  - расширения данных (уровень 4) трудны в реализации, так как описывают только синтаксис без механизмов описания семантики;
  - не стандартизован транспортный уровень передачи сообщений. Используется устаревший способ коммуникации точка-точка (RS232);
  - отсутствует стандартный прикладной программный интерфейс;
  - слишком много вопросов оставлено для решения при реализации (сложность).

WITSML – это прежде всего реализация WITS на современной языковой, транспортной и программной основе: через XML, Интернет и Web-сервисы. WITSML призван улучшить принятие решений, благодаря непрерывному потоку данных реального времени. Одно из его главных преимуществ – способность к легкому

расширению и дополнению для поддержки будущих технологий инженерии скважин. В то же время его принятие и внедрение способствует повышению уровня интероперабельности за счет продвижения отраслевых стандартов и увеличения доли XML-технологий. В основе WITSML лежит типичная для UDDI архитектура «Публикация / подписка». Опубликованные спецификации обменных форматов WITSML конкретного интегратора (заказчика) доступны постоянно. Подписавшийся клиент получает данные интегратора, к чьей опубликованной схеме он подключен. Чтобы извлекать данные предыстории, если потеряно подключение (например, перебои на линии спутниковой связи) с помощью не имеющих состояния Web-протоколов сервисы WITSML предусматривают специальный тип запроса «GetFromStore».

Поток данных WITS-0	Поток данных WITSML
&&	<?xml version="1.0" standalone="no" ?>
0201	<-logs xmlns="http://www.witsml.org/schemas/120"
02021	xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
0203152	xsi:schemaLocation="http://www.witsml.org/schemas/120
02041699	http://www.witsml.org/schemas/120/obj_log.xsd" version="1.2.0">
02270.973	<-log uidWell="W-B6CF4BDC94954A88AFDA298DED0446F1" uidWellbore=""
02288871	uidLog="L-9A8213E28ADF48B5941433F069CF5822"
0229484	<-nameWell /> <nameWellbore /> <nameLog />
!!	<-logHeader>
&&	<runNumber />
1401	<indexType>Depth</indexType>
14021	<startIndex>5342.12</startIndex>
14088871	<endIndex>5342.12</endIndex>
1419-9.25	<indexCurve>DEPTMEAS</indexCurve>
1420266	<stepIncrement>0</stepIncrement>
14211484	<indexUnits>FT</indexUnits>
	<logHeaderParam index="1" name="DATE" description="Log Date">11-May-2007</logHeaderParam>
	<-logCurveInfo>
	<mnemonic>DEPTMEAS</mnemonic>
	<unit>FT</unit>
	<nullValue>-999.2500</nullValue>
	...

Рис. 3. Пример потоков данных WITS и WITSML

«GetFromStore» позволяет не просто получать поток данных, но и выполнять прикладной запрос к доступным объектам, например, «Скважина», «Ствол скважины» и извлекать данные с сортировкой по мнемоникам, интервалам, обновлениям и т.п. Однако эта гибкость требует специальных усилий по обеспечению информационной безопасности. Основные данные промышленной геофизики могут быть получены только в процессе и непосредственно после бурения (до цементирования колонны). Поэтому стоят они дорого. По данным скважин можно оценивать запасы, а, следовательно, и стоимость месторождений. Это чрезвычайно ценная коммерческая информация. Чтобы ее обезопасить, сетевые ограничения, как правило, не позволяют входить в корпоративную сеть из внешней среды. Чтобы компании, занятые агрегацией данных, могли искать и проверять свои данные, непосредственно при бурении скважин им необходим удаленный доступ внутрь информационного центра сервисной компании, занятой бурением. В качестве альтернативы они могут предоставлять место в своей интрасети сервисным компаниям, однако в этом случае придется поддерживать множество систем разных сервисных компаний – со своей инфраструктурой взаимодействия и безопасности. Так или иначе, средой функционирования WITSML является интранет.

Документ WITSML представляется в удобном XML-формате. От людей, владеющих англоязычной терминологией промышленной геофизики, он не требует больших навыков для чтения с листа. А выше, на рис. 3 показано, что по сравнению с ним WITS, даже в текстовом виде уровня 0, – слишком «машинный», нечитабельный формат.

Помимо «неудобочитаемости» WITS создает проблемы интерпретации:

- любые данные необходимо записывать в каналы, что естественно для каротажных кривых, но неудобно для других типов информации;
- каналы WITS иногда сложно идентифицировать;
- некоторые типы кривых не определены в спецификациях WITS;
- назначение каналов WITS иногда изменяется по ошибке или в результате индивидуальной договоренности предприятий, нуждающихся в передаче дополнительных типов данных, и не имеющих такой возможности в рамках существующей жесткой спецификации.

По данным Petrolink [11] 95% проблем интеграции связаны со сбором и согласованием данных бурения скважин. WITSML очевидно может помочь улучшить ситуацию. Почему же WITS все еще не сдает позиции WITSML? Подобные вопросы возникают не только в промышленной геофизике. Почти во всех отраслях новые форматы на основе XML незначительно потеснили старые, несмотря на очевидные преимущества.

Причини жизнеспособности WITS, как типичного «старого» стандарта.

- WITS хорошо работает и прост в обработке. Зачем изменять то, что работает?
- Существующее оборудование поддерживает WITS. Стоит ли вкладывать в его модернизацию большие деньги ради формата?
- WITS-2b более надежен в условиях плохой связи: предыстория может быть восстановлена в случае обрыва линии из буфера. WITSML требует передачи пакета целиком.
- WITSML все еще плохо известен и более сложен для использования в полевой обработке.
- Гибкость WITSML в силу отсутствия общепризнанного индустриального образца потенциально чревата дублированием и путаницей в типах данных.

Последние два утверждения требуют пояснений. Теоретически, работа со схемой данных WITSML должна осуществляться достаточно просто, по взаимосвязям объектов: Скважина → Ствол скважины → Объект инклинометрии (траектории ствола) → Объект каротажа. Но каждая сервисная компания реализует эту схему по-своему. Одна хранит и каротажные данные, и информацию об инклинометрии в одном объекте, следовательно, стандартный запрос ML не будет срабатывать. Другая не следует обычной для WITS процедуре авторизации подключения типа точка-точка и т.п. В отличие от многократно растражированного и давно настроенного WITS практический опыт реализации и применения WITSML еще не устоялся.

Помимо перечисленных проблем общего характера распространению WITSML препятствуют специфические обстоятельства функционирования отрасли.

- WITSML был «изобретен» заказчиками, добывающими и крупнейшими сервисными компаниями, решавшими задачу сэкономить деньги на интеграции поступающих к ним данных от разных источников. Однако его внедрение зависит от сервисных операторов, непосредственно занятых бурением и сбором данных. Для них переход на новый формат чреват дополнительными расходами.

- Сервисные компании желают сохранить в своих руках управление собранными данными. Автоматизация контроля работ со стороны заказчика означает для них невозможность скрыть или как-то компенсировать неизбежные в любой работе ошибки.

- Данные нуждаются в качественном управлении и периодическом редактировании. Персонал, работающий непосредственно на скважинах, как правило, имеет меньше опыта первоначальной обработки и редактирования, чем персонал информационных центров сервисных компаний. Однако, последние обычно находятся вне системы сетевой защиты заказчика, и в случае замены протокола WITS точка-точка на интранет WITSML утратят права доступ к данным.

- У сервисных компаний могут быть частные данные и внутренние форматы, доступ к которым третьих лиц им нежелателен. Объединение с заказчиками в общую сеть интранет, необходимое для реализации безопасной сервис-ориентированной архитектуры WITSML, противоречит правилам внутренней безопасности многих компаний.

На сегодняшний день WITSML почти не используется на местах бурения скважин. И вероятно не будет использоваться, если этого не станут требовать заказчики (т.е. без включения этого требования в тендерные условия и контракты).

Реализация WITSML на местах бурения скважин потребует поддержки как минимум на уровне месторождения. Ограниченное группой скважин решение вряд ли имеет смысл для заказчика. При этом заказчику для решения своих задач требуется доступ к данным разных сервисных компаний, например, для контроля состояния и свойств бурового раствора. Однако использование бурового раствора – одно из «know-how», которым буровые сервисные компании не захотят делиться со своими конкурентами. В случае, если группа сервисных компаний, обслуживающих одно месторождение, для реализации WITSML должна будет войти в единую интрасеть, любые данные, доступные заказчику, станут доступны и остальным участникам этой интрасети. Кроме сервисных компаний непосредственно занятых работами на скважине в ситуацию конфликта интересов попадают обрабатывающие сервисные компании, так как они не хотят и не имеют права предоставлять доступ к частным данным своих заказчиков другим организациям.

Выгоды заказчика от реализации WITSML на местах бурения скважин тоже не нужно преувеличивать. Многочисленные сервисные компании не могут быть полностью совместимы с WITSML. Следовательно, потребуется настройка реализации запроса 'GetFromStore' для каждого нового варианта WITSML. Таким образом, современный гибкий формат, задуманный как замена устаревшего ограниченного WITS оказался не пригодным в качестве замены.

Тем не менее, затраты на разработку WITSML не оказались напрасными. Не заменив WITS во взаимодействии бизнесов, WITSML оказался весьма полезным средством межпрограммной и межсистемной интеграции, особенно в глобально распределенных компаниях. Вместо попарного связывания программ и корпоративных баз данных появилась возможность отображать все их внутренние схемы на схему WITSML, существенно экономя усилия на анализе и разработке. К настоящему времени на рынке появились готовые WITSML-адаптеры для ряда широко распространенных в мире геофизических программных систем, в том числе, Schlumberger, Halliburton, Paradigm.

Однако помимо этих систем в промышленной геофизике используются сотни менее популярных программных продуктов. Для большинства из них разработка и поддержка WITSML- или PRODML-адаптера – слишком дорогое удовольствие. Для интеграции таких продуктов посредством обмена XML-документами, в том числе, в стандартных форматах, прекрасно подходит механизм графовых запросов по образцу.

### Реализация графовых запросов по образцу для формирования XML-представлений

Сотрудниками Института кибернетики имени В.М. Глушкова НАН Украины и Украинского государственного геологоразведательного института разработан программный пакет «ГеоПоиск» [12] для интерпретации данных геологических исследований. При этом данные исследований импортируются в базу данных под управлением СУБД «МикроПоиск». Все приложения в составе системы «ГеоПоиск» используют эту БД (ее фрагмент на рис. 4.) в качестве источника данных. Для доступа к данным используется встроенный язык запросов.

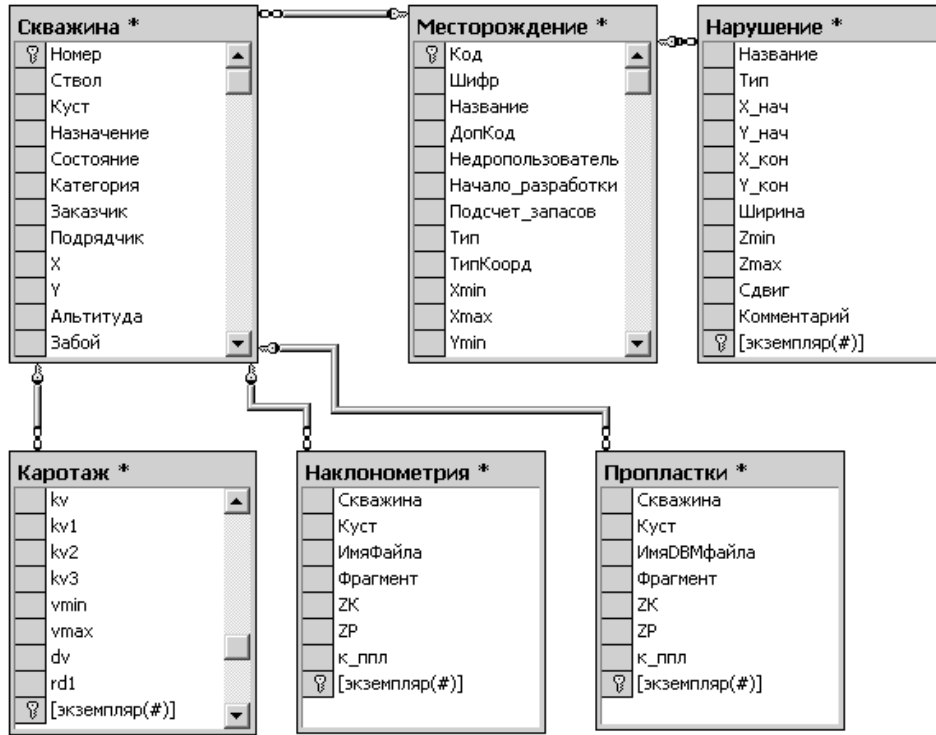


Рис. 4. ER-диаграмма фрагмента БД «ГеоПоиск»

Для целей просмотра и экспорта данных из системы «ГеоПоиск» создано специальное QBE-приложение «Селектор» [13]. В нем можно подготовить данные для экспорта, составляя дерево из объектов БД. Для каждого объекта можно выбрать список экспортируемых атрибутов, на каждый атрибут можно наложить условие (фильтр) (см. дерево на рис. 5).

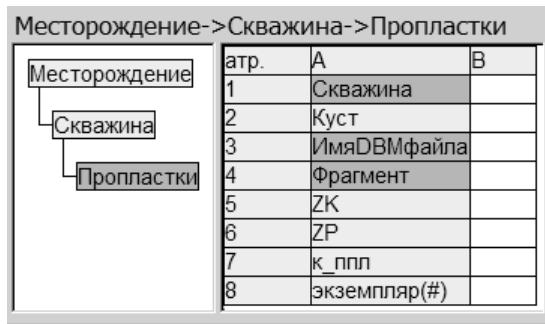


Рис. 5. Пример графового запроса в «Селекторе»

Дерево для экспорта получается путем обхода объектов схемы базы данных произвольным маршрутом. При этом схема данных рассматривается как неориентированный граф. Это позволяет рассматривать отношения 1:N между объектами двояко: как переход от 1 к N, либо как 1 к 1. Например, возможны переходы Месторождения → Скважина и Скважина → Месторождение. После того, как данные подготовлены, они могут быть экспортированы в виде отформатированных текстовых файлов. Для каждого узла дерева экспорта можно создать отдельную текстовую таблицу.

Такой подход обладает рядом недостатков. В частности, данные из полученного набора текстовых файлов будут разрозненные, не связанные между собой. Иерархическая связь между объектами будет потеряна. Для устранения этого недостатка можно использовать XML как формат экспортируемых данных. Использование XML позволяет:

- а) экспортировать иерархию объектов вместе со значениями их атрибутов;
- в) применять средства XQuery для преобразования полученных данных.

При экспорте в формат XML каждый атрибут рассматривается как скалярный элемент, а каждый объект – как элемент, содержащий другие элементы.

В работах [14, 15] предложены графовые запросы в качестве средства формирования XML-взглядов для реляционных СУБД. Доказано, что они обладают полнотой для представления сетевой концептуальной модели посредством иерархии взглядов. Для визуального формирования графовых запросов использован QBE-подобный язык ER-QBE и реализована его трансляция в SQL-подобный язык. Реализацией этого языка является «Селектор».

**Применение XQuery для объединения XML-представлений.** XQuery – это язык получения и обработки данных из XML-документов. В этом языке используются выражения XPath для доступа к элементам XML-документа, а так же FLWOR-выражения (аббревиатура содержит имена операций: For, Let, Where, Order by, Return) для обеспечения возможности объединения нескольких документов и агрегации данных. В начале 2007 года XQuery рекомендован консорциумом W3C для преобразования XML-данных в Интернет-приложениях [16].

Рассмотрим использование XQuery для сведения к унифицированной форме данных из различных БД на примере XML-документов, созданных в пакетах «ГеоПоиск» и «Tesseral Pro». Экспорт этих XML-документов выполняет программа «Селектор». Из «ГеоПоиска» экспортированы агрегированные данные по пропласткам, из «Tesseral Pro» – данные каротажа. На рис. 6 показаны фрагменты этих XML-документов (после выполнения этапа унификации однотипных имен объектов, который был опущен для краткости).

<pre> &lt;?xml version="1.0" standalone="yes" ?&gt; - &lt;xmlDB&gt; + &lt;Mestorozdenie&gt; - &lt;Mestorozdenie&gt;   &lt;ID&gt;2&lt;/ID&gt;   &lt;Shifr&gt;0&lt;/Shifr&gt;   &lt;Name&gt;Tehnologicheskoe&lt;/Name&gt; + &lt;Skvazyna&gt; - &lt;Skvazyna&gt;   &lt;ID&gt;40&lt;/ID&gt;   &lt;Nomer&gt;3001&lt;/Nomer&gt;   &lt;X&gt;0&lt;/X&gt;   &lt;Y&gt;0&lt;/Y&gt; - &lt;Karotaz&gt;   &lt;ID&gt;347&lt;/ID&gt;   &lt;krovlja&gt;2892.2&lt;/krovlja&gt;   &lt;podoshva&gt;3067.8&lt;/podoshva&gt;   &lt;/Karotaz&gt; + &lt;Karotaz&gt;   &lt;/Skvazyna&gt; + &lt;Skvazyna&gt;   &lt;/Mestorozdenie&gt; &lt;/xmlDB&gt;                 </pre> <p style="text-align: center;">a</p>	<pre> &lt;?xml version="1.0" standalone="yes" ?&gt; - &lt;xmlDB&gt; + &lt;Mestorozdenie&gt; - &lt;Mestorozdenie&gt;   &lt;ID&gt;2&lt;/ID&gt;   &lt;Shifr&gt;0&lt;/Shifr&gt;   &lt;Name&gt;Tehnologicheskoe&lt;/Name&gt; + &lt;Skvazyna&gt; - &lt;Skvazyna&gt;   &lt;ID&gt;40&lt;/ID&gt;   &lt;Nomer&gt;3001&lt;/Nomer&gt;   &lt;X&gt;0&lt;/X&gt;   &lt;Y&gt;0&lt;/Y&gt; - &lt;Proplastki&gt;   &lt;ID&gt;25&lt;/ID&gt;   &lt;Fragment&gt;0&lt;/Fragment&gt;   &lt;Skvazyna&gt;301&lt;/Skvazyna&gt;   &lt;/Proplastki&gt;   &lt;/Skvazyna&gt; + &lt;Skvazyna&gt;   &lt;/Mestorozdenie&gt; &lt;/xmlDB&gt;                 </pre> <p style="text-align: center;">б</p>
--	---

Рис. 6. XML-документы для БД:

а) Месторождение – Скважина – Каротаж; б) Месторождение – Скважина – Пропластки

На рис.7. показан XQuery-запрос для объединения двух XML-документов.

<pre> &lt;xmlDB&gt; {for \$a in doc("karotaz.xml")/xmlDB/Mestorozdenie for \$b in doc("proplastki.xml")/xmlDB/Mestorozdenie where \$b/ID = \$a/ID return &lt;Mestorozdenie&gt; {   &lt;ID&gt;{\$a/ID/text()}&lt;/ID&gt;,   &lt;Shifr&gt;{\$a/Shifr/text()}&lt;/Shifr&gt;,   &lt;Name&gt;{\$a/Name/text()}&lt;/Name&gt;,   for \$d in \$a/Skvazyna   for \$e in \$b/Skvazyna   where (\$d/ID = \$e/ID)   return &lt;Skvazyna&gt;   {     &lt;ID&gt;{\$d/ID/text()}&lt;/ID&gt;,     &lt;Nomer&gt;{\$d/Nomer/text()}&lt;/Nomer&gt;,     &lt;X&gt;{\$d/X/text()}&lt;/X&gt;,     &lt;Y&gt;{\$d/Y/text()}&lt;/Y&gt;,     for \$f in \$d/Karotaz     where (count(\$f) &gt; 0)     return \$f,     for \$g in \$e/Proplastki     where (count(\$g) &gt; 0)     return \$g   }&lt;/Skvazyna&gt; }&lt;/Mestorozdenie&gt; } &lt;/xmlDB&gt;                 </pre> <p style="text-align: center;">a</p>	<pre> &lt;?xml version="1.0" encoding="UTF-8" ?&gt; - &lt;xmlDB&gt; + &lt;Mestorozdenie&gt; - &lt;Mestorozdenie&gt;   &lt;ID&gt;2&lt;/ID&gt;   &lt;Shifr&gt;0&lt;/Shifr&gt;   &lt;Name&gt;Tehnologicheskoe&lt;/Name&gt; + &lt;Skvazyna&gt; - &lt;Skvazyna&gt;   &lt;ID&gt;40&lt;/ID&gt;   &lt;Nomer&gt;3001&lt;/Nomer&gt;   &lt;X&gt;0&lt;/X&gt;   &lt;Y&gt;0&lt;/Y&gt; - &lt;Karotaz&gt;   &lt;ID&gt;347&lt;/ID&gt;   &lt;krovlja&gt;2892.2&lt;/krovlja&gt;   &lt;podoshva&gt;3067.8&lt;/podoshva&gt;   &lt;/Karotaz&gt; + &lt;Karotaz&gt; - &lt;Proplastki&gt;   &lt;ID&gt;25&lt;/ID&gt;   &lt;Fragment&gt;0&lt;/Fragment&gt;   &lt;Skvazyna&gt;301&lt;/Skvazyna&gt;   &lt;/Proplastki&gt;   &lt;/Skvazyna&gt; + &lt;Skvazyna&gt;   &lt;/Mestorozdenie&gt; &lt;/xmlDB&gt;                 </pre> <p style="text-align: center;">б</p>
--	---

Рис. 7. Интеграция данных систем ГеоПоиск и TesseralPro:

а) XQuery-запрос, б) Результат интеграции XML-документов

Как видно из рис. 7, б, в полученном документе интегрированы данные двух программных систем.

Язык XML – эффективное средство интеграции данных, в том числе для применения в Интернет-приложениях и информационных хранилищах. Разработанный QBE-подобный язык графовых запросов и его реализация в приложении «Селектор» применяются для обмена данными между пакетами «ГеоПоиск» и «Tesseral Pro» [17] и являются полезными компонентами для создания и поддержки корпоративных баз данных и информационных хранилищ геофизических организаций. Открытым пока остается вопрос обновления баз данных на основе интегрированных XML-документов. Однако развитие средств импорта XML в большинстве коммерческих СУБД, а также принятие XML в качестве внутреннего формата нового MS Office 2007, будет способствовать быстрому разрешению этой проблемы.

## **Заключение**

В силу практической невозможности повторения некоторых геофизических исследований после цементирования колонны скважины и высокой цены воспроизведения других типов исследований в геофизических и нефтедобывающих компаниях принято хранить все исходные данные и варианты их интерпретации. С появлением новой информации возникает возможность уточнить результаты исследований по старым данным. Накопленные данные хранятся в разных, зачастую устаревших форматах от производителей измерительных устройств и в базах данных программных пакетов. Это приводит к целесообразности создания информационного хранилища геофизических данных. Под информационным хранилищем понимают единую, полноценную, систематизированную и непротиворечивую совокупность данных, извлеченных из различных источников и ставших доступными конечным пользователям для понимания и применения [18]. Интеграция данных – необходимый этап создания и поддержки информационных хранилищ.

1. *Sheth, A. P.* (1999). Changing focus on interoperability in information systems: from system, syntax, structure to semantics. In M. Goodchild, M. Egenhofer, R. Fegeas, and C. Kottman (Eds.), *Interoperating Geographic Information Systems*. Kluwer, P. 5–29.
2. *International Stratigraphic Chart* - <http://www.stratigraphy.org/cheu.pdf>
3. *Wellsite Information Transfer Specification* - <http://home.sprynet.com/~carob>
4. *LAS – Log ASCII Standard* - [http://www.cwls.org/las\\_info.php](http://www.cwls.org/las_info.php)
5. *WITSML – Wellsite Information Transfer Standard Mark-up Language* (<http://www.witsml.net/main>)
6. *PRODML – Production Markup Language* - <http://www.prodml.org>
7. *XMML – eXploration and Mining Markup Language* (<https://www.seegrid.csiro.au/twiki/bin/view/Xmml/WebHome>)
8. *GeoSciML – Markup Language for the Geosciences* (<http://www.geosciml.org>)
9. *Geography Markup Language* (<http://www.opengeospatial.org/standards/gml>)
10. *Bray T., Paoli J., Sperberg-McQueen C.M.* Extensible Markup Language (XML) 1.0 (2nd Edition). – W3C Recommendation, 2000. – 771 p.
11. [http://www.witsml.org/images/poc/meetings/may07witpub/may07witpub\\_22\\_petrolink.ppt](http://www.witsml.org/images/poc/meetings/may07witpub/may07witpub_22_petrolink.ppt)
12. [www.geopoisk.com](http://www.geopoisk.com)
13. *Гречко В.О., Тульчинский В.Г., Тульчинский П.Г.* От QBE к графовым запросам //Труды VI Междунар. конф. "Знание–Диалог–Решение". – Киев: Ассоциация создателей и пользователей интеллектуальных машин. – 1997. – Том 1. – С. 216–224.
14. *Tulchinsky V.G., Iushchenko O.K., Iushchenko R.A.* Improving Data Integration by Means of XML – 71th EAGE Conference & Exhibition incorporating SPE EUROPEC 2009. – Amsterdam, Netherlands. – 2008, #P056, <http://www.earthdoc.org/detail.php?pubid=23702>
15. *Тульчинский В.Г., Юценко А.К., Юценко Р.А.* Графовые запросы для интеграции данных посредством XML // Кибернетика и Системный Анализ. – 2008. – № 2. – С. 171–183. - <http://www.springerlink.com/content/t468k03016576v68/fulltext.pdf>
16. *Chamberlin, D., D. Florescu, J. Robie, J. Simeon, and M. Stefanescu.* XQuery: a query language for XML, 2001 (<http://www.w3.org/TR/query>)
17. [www.tesseral-geo.com](http://www.tesseral-geo.com)
18. *Гречко В.О., Тульчинский В.Г., Тульчинский П.Г., Харченко А.В.* Информационные хранилища в промышленной геофизике // Проблемы программирования. – 2000. – № 1, 2. – С. 81 – 90.

**The application of XML-representation for data integration in petroleum geophysics,  
Tulchinsky Vadim, Iushchenko Oleksandra, Iushchenko Ruslan.**