Российская академия наук
Сибирское отделение
Институт систем информатики
им. А. П. Ершова

А. Г. Марчук

РАСПРЕДЕЛЕННЫЕ ЭЛЕКТРОННЫЕ АРХИВЫ, БИБЛИОТЕКИ И БАЗЫ ДАННЫХ

Препринт
122

Новосибирск 2004

В работе описываются принципы, архитектура и программная реализация подхода, позволяющего создавать распределенные информационные системы класса Semantic Web [1]. Кроме использования идей и технологий Semantic Web, данный подход основывается на уточненном представлении об информационном пространстве и на выработанной автором схеме данных, позволяющей отражать сущности и связи реального мира. Модель является открытой и расширяемой, что дает возможность создавать широкий класс универсальных и специализированных систем.

Siberian Division of the Russian Academy of Sciences
A. P. Ershov Institute of Informatics Systems

Alexander G. Marchuk

DISTRIBUTED ELECTRONIC ARCHIVES, LIBRARIES AND DATA BASES

Preprint
122

Novosibirsk 2004

This preprint describes the concepts, architecture and implementation of our approach to creation of information systems of the semantic-web class [1]. In addition to the ideas and technology of semantic web, this approach is based on the refined notion of the information space and on the data scheme elaborated by the author for representation of real-world entities and relationships. The model is open and extendable, which allows a wide class of universal and special-purpose systems to be created.

ВВЕДЕНИЕ

Прошедшая в 90-х и начале 2000-х годах реформация базовых решений по работе с данными (XML, RDF и др.), построению клиент-серверных приложений (Web, CORBA и др.), средств и платформ объектно-ориентированного программирования и компонентного проектирования (Java, .NET и др.), создает предпосылки для ревизии представлений об архитектуре информационных систем. Другим заметным фактором является моральное устаревание систем, основанных на реляционных СУБД (Oracle) и технологиях построения систем, основанных на хранилищах данных. При этом, использование новых подходов плохо сочетается с традиционными технологиями и требует построения новой методологии, а за ней платформы и ее окружения.

Проблема, требующая своего решения, состоит в представлении данных в виде централизованных образований, не способных к естественным формам объединения. Поэтому, владелец информационного ресурса стоит перед дилеммой: либо «отдать» свои данные в «чужие руки» и потерять над ними контроль, либо полностью автономизироваться и в своей базе данных поддерживать не только данные своей специфики, но и сопряженные с ними неспецифические данные, необходимые ему для работы, а значит — дублировать информацию, имеющуюся в (многочисленных) других источниках. Часто оба варианта являются неприемлемыми. Это наиболее существенно в задачах построения архивных и музейных систем, в простых системах делопроизводства.

В данной работе будет показано, что наиболее естественно строить такие системы в технологии двухуровневых построений по данным и через накопление базы фактов в совокупности образующей модель мира или ее фрагмент.

КОНЦЕПЦИЯ

Для формулирования концепции рассмотрим две модели.

В левой части рис. 1 изображены данные и программы, их обрабатывающие. Если рассмотреть традиционную пару «данные—программы», то можно задать вопрос: где находится смысл обрабатываемых данных? Ответ

* Эта работа частично поддержана РФФИ, грант № 03-07-90330.

очевиден — смысл данных «знают» программы обработки, именно в их коде «зашито» понимание этих данных. В правой части рисунка появляются метаданные, т.е. данные о данных. В метаданные мы можем попытаться вложить смысл данных или часть смысла. Если метаданные обрабатывать вместе с данными, то программы смогут стать универсальными, что явно выглядит как прогрессивный фактор.

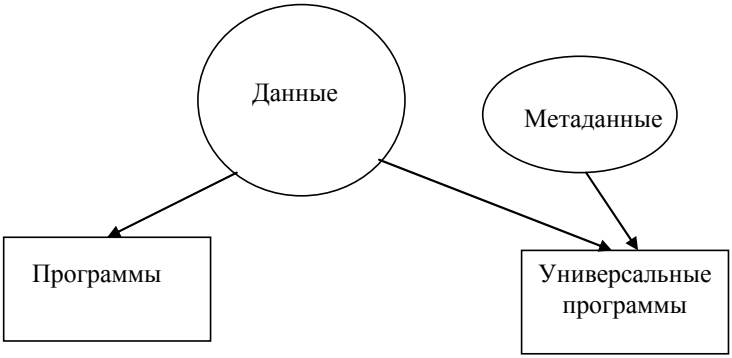

Рис. 1

На рис. 2 изображены различные информационные системы ИС 1, ИС 2,…, работающие каждая со своей системой данных. Однако, как легко видеть, практически каждая информационная система, сколь бы специализированной она не была, в свою обработку вовлекает данные, являющиеся отражением традиционных сущностей реального мира, таких как персоны, организации, события, адреса и т.д. Назовем эти данные неспецифическими в противовес специфическим для каждой ИС системы «своих» данных, отражающих предметные особенности информационной системы. Правильным построением информационных систем в соответствии с данной моделью является двухуровневое разбиение данных на общие (неспецифические) и локальные (специфические) данные, как это изображено на рис. 2. Проблема заключается в том, что традиционные технологии не позволяют, точнее, не помогают, реализовывать такую архитектуру. Двухуровневость разбиения данных достаточно условна. Понятно, что в развитых случаях обобществлению могут подвергаться и специфические данные, а в целом, общее поле данных может иметь некоторую структуру, как минимум — оно распределенное.

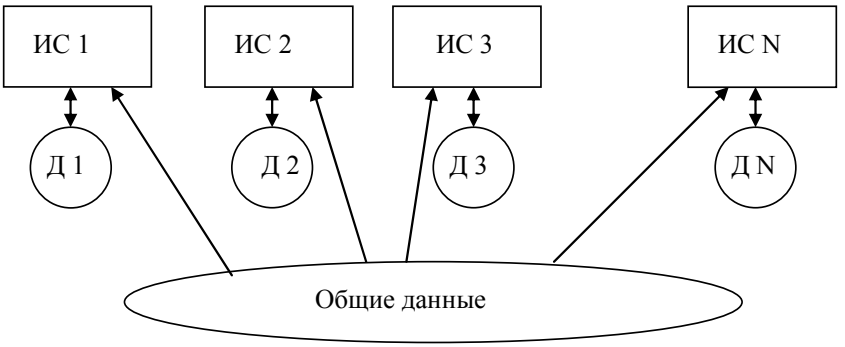

Рис. 2

Информационные системы нового поколения видятся как распределенные системы, опирающиеся на множественные базы «осмысленных» данных, содержащие неспецифические общие данные, неспецифические приватные данные, специфические для информационной системы общие и частные данные, спецификации модели мира, предметной области и задачи. Совместимость данных и описаний должна обеспечиваться общей методологией, едиными стандартами. Программный код таких систем должен быть универсальным, интерпретирующим произвольные спецификации и данные, или специализированным, сгенерированным в стиле смешанных вычислений из универсального решения, фиксацией нужного контекста.

Наиболее адекватной парадигмой для построения таких систем является подход Semantic Web [2], сформированный консорциумом W3C, разрабатывающим также соответствующие стандарты, такие как RDF, RDFS, OWL [3, 4, 5]. База данных при таком подходе формируется из (распределенного) множества семантических сетей, построенных по простым правилам, включающих в себя данные в виде высказываний, «склеивающихся» в единый граф через склейку по уникальным идентификаторам сущностей и разделенных по предмету и источнику через RDF-документы и пространства имен. Семантическими сетями описываются данные, метаданные, схемы данных, другие структурные построения, требующиеся для целостного описания информационной системы. Если технологическое решение, позволяющее объединять данные из разных источников, уже имеется (RDF), то единого подхода для структурирования фактов реального мира (модели мира) пока не создано. Также непроработанным является вопрос спецификации функционирования информационной системы и интерфейсов.

Для более четкого формирования и обоснования принципов и архитектуры информационных систем, построенных на общих данных, рассмотрим уточнение представления об информационном пространстве.

ИНФОРМАЦИОННОЕ ПРОСТРАНСТВО ДОКУМЕНТОВ

Данный раздел развивает подход, описанный в работе автора [6]. Под пространством документов будем представлять интуитивно понятную среду существования опубликованных информационных материалов. Публикации понимаются как классические — в книгах, журналах и др. печатных изданиях, так и современные — Интернет-публикации, документы в файлах. Экземпляром документа (ЭД) назовем конкретный экземпляр публикации (файл, экземпляр книги и др.). Первичным в модели пространства будем считать множество экземпляров.

Рассмотрим самые общие свойства такого пространства. Во-первых, пространство не является статичным. В общем случае, это означает, что экземпляры документов (в пространстве) появляются, исчезают и изменяются. Во-вторых, экземпляр документа — это его содержимое (content) и его координата. Таким образом, пространство предполагается координатным, причем все координаты — уникальны, т.е. одновременно не существует двух разных экземпляров документов, занимающих одну координату. Содержимое экземпляра документа можно рассматривать как информацию, хранящуюся в документе в виде набора байтов. Для электронных публикаций содержимое, как правило, совпадает с набором байтов файла этой публикации.

В рамках такой модели уже можно говорить об одинаковых экземплярах документов, если их содержимое совпадает. Также очевидным образом вводятся понятия копирования и перемещения ЭД.

Следующий уровень абстракции связан с изменяемостью экземпляров документов. Допущение изменяемости ЭД требует введение нового понятия — понятия документа. Здесь надо снова опереться на интуитивное представление о документе, как о носителе информации для каких-то определенных целей. На интуитивном уровне изменение документа это его «улучшение», типа «13-е издание, исправленное и дополненное». Формально, определим документ как некоторое подмножество экземпляров документа, в котором есть выделенный экземпляр, называемый оригиналом. Понятно, что в данном подмножестве также могут быть одинаковые экземпляры, а факторизацию этого подмножества по эквивалентности естествен-

но назвать версиями документа. Последняя версия документа совпадает с множеством одинаковых ЭД, среди которых есть оригинал.

Следующее развитие модели информационного пространства документов связано с введением идентификации и отметок времени. Сопоставим каждому документу пространства уникальный идентификатор, а каждому экземпляру документа — идентификатор его документа и время последнего изменения экземпляра. Теперь будем предполагать, что ЭД — это четверка: идентификатор (документа), координата, содержимое и отметка времени последнего изменения. Соответственно, целесообразно запретить изменение документа без изменения отметки времени, а более позднее значение временной отметки должно соответствовать более позднему, в смысле последовательности событий, изменению ЭД. Практика работы с документами позволяет также считать, что изменения должны производиться только с последней версией, хотя в некоторых случаях это не так, например, при «откате» к предыдущим версиям.

Такое введение новых сущностей в понятие документа и экземпляра документа дает возможность изменить недостаточно конструктивное понятие совпадения ЭД на конструктивное и легко проверяемое понятие эквивалентности. Два ЭД эквивалентны, если совпадают их идентификаторы и временные отметки. Совпадение содержимого при этом будет гарантироваться, если:

1) копирование и перемещение осуществляются с воспроизведением всех составляющих ЭД, кроме координаты;
2) при других процессах, изменяющих содержимое экземпляра документа, метка времени не должна совпадать с уже имеющимися для данного документа.

Получающаяся модель пока разделяет создаваемое информационное пространство на два: собственно пространство документов, состоящее из опубликованных ЭД, и пространства для хранения метаинформации, т.е. информации о документах и их экземплярах. Часто такое разделение сохраняют и концептуально. При этом, метаинформацию коллекционируют в реестрах — выделенных информационных объектах. Наша цель — объединить обе сущности в едином, специально организованном, информационном пространстве.

Рассмотрим структуру метаинформации для предложенной модели. Из предыдущего следует, что каждому ЭД целесообразно сопоставить метаинформационную единицу, имеющую поля: идентификатор ЭД, идентификатор документа, координату содержимого и отметку времени последнего изменения. Также, каждому документу сопоставим аналогичную единицу с

полями: идентификатор документа, идентификатор ЭД оригинала. Будем пока предполагать, что вся эта метаинформация доступна из некоторого абстрактного реестра.

Рассмотрим использование (не изменение!) документа каким-то агентом. Исходным для процесса доступа является уникальный идентификатор документа. Требуется получить координату ЭД, эквивалентного оригиналу документа. Последовательность действий для этого достаточно простая: по идентификатору документа получаем идентификатор ЭД оригинала, а по нему получаем координату содержимого. Если доступ к удаленному оригиналу «дороже» доступа к более близкой копии, например из-за размеров оригинала или ограничений на трафик, то последовательность изменяется.

1. По идентификатору документа получаем идентификатор ЭД оригинала.
2. По идентификатору ЭД оригинала получаем временную отметку оригинала.
3. По идентификатору документа (ИД) и временной отметке (ВО) среди экземпляров документа, удовлетворяющих совпадению ИД и ВО, находим «ближнее» содержимое для использования.

Очевидно, нахождение «ближнего» является эвристическим процессом, основанным на оценке «стоимости» перекачки данных содержимого, а эта оценка должна учитывать ряд факторов, включая факторы общей оптимизации производительности информационной системы. Например, в простом случае, в модели доступа, областями информационного пространства будут: Интернет, Интранет, локальный диск сервера. В более сложной модели могут участвовать доступы к оптическим и другим внешним устройствам, размещение в оперативной памяти и др.

Предложенная модель ориентируется на два прагматических предположения: экземпляры документов не исчезают (или не «портятся») слишком быстро и, второе, если за время доступа к содержимому оригинал документа успел измениться, пользователь, тем не менее, получил осмысленную услугу. Первое требование/предположение может быть снято, если экземпляры документов являются статическими, хотя в практическом плане это полагать нереально. Ко второй части комментарий состоит в том, что пользователь мог бы «чуть раньше» или «чуть быстрее» осуществить свой доступ, так что относительность времени надо учитывать при построении архитектуры конкретных информационных систем, базирующихся на едином информационном пространстве.

Теперь рассмотрим динамику изменения метаданных, т.е. данных реестра. Поскольку мы не предполагаем возможность одновременного изменения целой группы, возможно разнесенных в пространстве, данных и

ния целой группы, возможно разнесенных в пространстве, данных и метаданных, то изменения должны производится в строгой последовательности. Рассмотрим следующие изменения: изменение содержимого ЭД, появление документа и его ЭД, изменение документа через появление нового ЭД.

Простейшее действие — изменение содержимого экземпляра документа. Последовательно выполняются два действия: изменяется содержимое и вносится новая временная отметка в запись о данном ЭД. В промежутке между этими действиями внешние пользователи могут получить новое содержимое под старой метаинформацией, но в большинстве случаев в этом проблемы нет.

Появление экземпляра документа происходит через публикацию содержимого ЭД с последующим появлением в реестре метаинформации. Проблема заключается в том, какой идентификатор документа прописать в поле записи об ЭД. Здесь есть два варианта: экземпляр документа является новым для уже существующего документа (изменение документа) и, второй, формируется новый документ. В первом случае вопрос разрешается очевидным образом, т.е. именно этот идентификатор записывается, а затем изменяется ссылка на оригинал, имеющаяся в записи о документе. Во втором случае перед фиксацией записи об ЭД заводится идентификатор нового документа, он записывается в соответствующее поле ЭД. В этот момент ЭД еще не может быть использован, потому что на него нет ссылки из документной части реестра. После фиксации нового ЭД в реестре, формируется запись о появившемся новом документе и она записывается в реестр. Описанный алгоритм соответствует варианту появления документа.

Теперь конкретизируем представление о реестре. Будем исходить из децентрализованной модели реестра. Это означает, что реестровая метаинформация распределена по опубликованным документам.

Пусть из документов специального вида можно извлекать записанную там метаинформацию о документах и экземплярах документов. И пусть какое-то количество реестровых записей извлечено. Сразу предположим, что мы не имеем полного реестра для текущей обработки. Рассмотрим каноническую операцию преобразования идентификатора документа в координату экземпляра, являющегося копией оригинала. Некорректным, но иногда приемлемым решением будет проверить имеющиеся в текущем реестре записи ЭД на совпадение с идентификатором документа и выбрать «наилучший». Однако для корректного разрешения задачи, мы должны иметь доступ к дескриптору документа и дескриптору ЭД оригинала. Тогда имеется информация о временной отметке оригинала и, если нет подходящей копии, можно воспользоваться ссылкой на ЭД оригинала.

Проблема состоит в том, что этих двух дескрипторов может не оказаться в рабочей области. Задачу мы сможем решить, если суметь организовать детерминированный процесс нахождения в нашем пространстве документа, хранящего регистрационную запись, соответствующую любому из применяемых для документов идентификаторов. Общая схема решения данной задачи состоит в том, что мы будем предполагать доменную структуру идентификаторов документов. Это означает, что идентификатор состоит из цепочки локальных идентификаторов. Тогда если положить, что упоминавшийся «документ, хранящий регистрационную запись», идентифицирован в домене более высокого уровня по иерархии, то по индукции можно доказать, что при небольших предположениях на начальную стадию индукции, существует процесс его нахождения.

Рассмотрим предлагаемую схему подробнее. Пусть есть идентификатор документа Id и гипотетическая функция Translate(Id), осуществляющая преобразование идентификатора документа в координату его оригинала.

Как мы уже предположили, метаинформация записана в документах этого же пространства, поэтому для чтения полей метазаписи нужен экземпляр соответствующего документа. Для извлечения метазаписей из документов нам понадобится функция ExtractMetarecord(Dcoord, Id), извлекающая метазапись, соответствующую идентификатору Id некоторой сущности из документа с координатой Dcoord. Функция осмысленна, только если документ содержит эту метазапись. Теперь

$$\text{Translate(Id)} = \text{ExtractMetarecord(Ecoord, DE).Coord},$$

где DE — идентификатор оригинала документа, а Ecoord — координата документа-реестра.

$$\text{DE} = \text{ExtractMetarecord(Dcoord, Id).Original}.$$

Для упрощения, предположим, что метазаписи документа и его оригинала находятся в одном документе. Теперь, чтобы решить задачу, надо предположить возможность определения идентификатора этого документа по имеющейся информации. Этому помогает доменная организация идентификаторов. Действительно, если имеющийся идентификатор Id «укоротить», то можно потребовать, чтобы это был идентификатор документа (директории), содержащего ВСЕ метазаписи объектов со структурой {идентификатор директории}/{короткий идентификатор объекта в данной директории}. Введя соответствующую функцию Dir(Id), лексически вычисляющую идентификатор директории, получим решение нашей задачи:

Translate(Id) = ExtractMetarecord(Translate(Dir(Id)),
ExtractMetarecord(Translate(Dir(Id)), Id).Original).Coord.

Если рангом идентификатора назвать длину цепочки составляющих его имен, то рекурсия функции понижает этот ранг и некоторая другая реализация начальной системы директорий успешно завершает рекурсию. Кроме того, из структуры функции очевидно, что процесс вычисления не ветвится, а развивается линейно относительно структуры доменного имени.

Легко избавиться от ограничения, связанного с единственностью документа, определяющего содержимое директории. В этом случае, объект, помеченный идентификатором Dir(Id), представляет собой более сложную конструкцию, но для наших целей детали устройства не важны, и достаточно изменений, описанных ниже.

Введем функцию DirDocSet(Id), выдающую по идентификатору документа множество координат документов, в совокупности являющихся реестром данной директории. А функцию ExtractMetarecord будем теперь трактовать как имеющую первый аргумент — множество координат. Изменение структуры реализации функции Translate очевидно.

Важно заметить, что данное построение информационного пространства документов не требует, чтобы метаинформационные записи целостно «хранились» в (документах) директории, а это важно для реализации принципов Semantic Web. Но «критические» поля дескрипторов (ссылка на оригинал для документа и координата для экземпляра документа) должны присутствовать среди директорных данных.

Теперь вернемся к экземплярам документов и использованию копий вместо оригиналов. В модели для этого также потребуются некоторые изменения. Во-первых, теперь будем трактовать Translate как функцию, выдающую координату «лучшего» экземпляра документа, эквивалентного оригиналу. Понятие «лучший» скроем функцией Best(Coords), где Coords — множество координат ЭД, эквивалентных оригиналу. Результатом функции является одна координата (лучшая, по мнению критериев программы). Другой новой функцией является EquivCoords(MR), где аргумент — метазапись экземпляра документа, а результат функции — множество координат ЭД, эквивалентных ЭД аргумента.

Теперь уточним структуру функции Translate, выдающей по идентификатору документа координату «ближнего» экземпляра документа. Чтобы не дублировать общие вычисления, как в предыдущем случае, разделим определение функции на два:

```
RegistryCoords = Translate(DirDocSet(Id)),
Translate(Id) = Best(EquivCoords(ExtractMetarecord(RegistryCoords,
        ExtractMetarecord(RegistryCoords, Id).Original))).
```

Во внутренних вызовах предполагается, что функция Translate применяется к множественным аргументам покомпонентно с получением множества значений.

ИНФОРМАЦИОННАЯ МОДЕЛЬ

Информационное пространство предназначено для отображения данных и метаданных. Разграничение между данными и метаданными достаточно условно, в силу того что любые данные, по сути, есть отражение некоторых сущностей реального или идеального миров. Если под идеальным миром понимать мир знаков и информации, то метаданные «обслуживают» именно его.

Базовым понятием для формирования информационного пространства является понятие сущности. Под сущностью мы понимаем нечто, о чем можно делать высказывания. Причем, оставаясь в материалистическом видении мира, высказывания определенной категории — факты. Это простые факты, не требующие сложных доказательств и не отражающие субъективную точку зрения. Например, то, что некая персона имеет конкретную дату рождения, фиксируемую в записи об этой персоне, является простым, хотя иногда сложно доказуемым фактом, а то, хороший это или плохой человек, выдающийся он или нет и т.д., фактом не является и прямым образом не должно фиксироваться в информационном пространстве. Косвенно это можно сделать через документы, т.е. документы, как реально существующие объекты, могут присутствовать в данных информационного пространства. Документ может отражать что-то, например человека, и это тоже является фактом. При этом документ может представлять субъективную позицию автора, в частности, в форме оценок.

Важным свойством отражаемых сущностей является различимость. Мы в состоянии определить, относятся ли две группы высказываний к одной сущности или разным. В первом случае мы говорим об отождествлении сущностей. Сама проблема отождествления в практике работы с информацией не является тривиальной, и ее особенности мы рассмотрим в следующих работах.

Различимость порождает возможность отражения сущностей через идентификацию. Это означает, что отражаемым сущностям мы сопоставляем уникальные идентификаторы, и эти уникальные идентификаторы становятся (в информационном пространстве) их заместителями (или знаками-заместителями). А возможность делать высказывания (устанавливать факты) реализуется через установление отношений между сущностями. В «чистом» виде можно говорить об одноместных, двухместных и многоместных отношениях. В «реальной» ситуации отношения также могут быть (стать) сущностями, поскольку их можно различать и о них можно делать высказывания. Например, отношение между персоной и работодателем может подвергаться дополнительному анализу (какая должность, с какой и по какую дату и др.).

Фиксация высказываний (фактов) в информационном пространстве может осуществляться с помощью разных формализмов. Это же касается формализмов идентификации и отождествления. Не останавливаясь на обосновании, остановим выбор на формализме RDF (Resource Description Framework) [3]. В этом формализме идентификация осуществляется через использование пространств имен и URI (Universal Resource Identifier) [7], а отношение представляет собой предикат между субъектом и объектом. Отождествление в RDF осуществляется путем «склейки» всех высказываний, относящихся к сущности, помеченной одним идентификатором. Отметим сразу, что это не решает всех проблем отождествления, поэтому требуется еще дополнительный механизм установления отношения тождественности между узлами, помеченными разными идентификаторами. Сложные отношения могут конструироваться в RDF порождением специальных сущностей и установления связей простыми предикатами.

Альтернативами данному формализму могли бы быть Пролог, реляционные базы данных или даже системы типов или классов в языках программирования.

БАЗОВАЯ АРХИТЕКТУРА

Снова вернемся к информационному пространству документов. Ограничивая себя структуризацией RDF, мы будем предполагать, что элементами (точками) пространства являются произвольные документы и документы RDF. Документы RDF являются основными носителями информации и метаинформации и порождают семантическую сеть, другие же документы — это всего лишь конечные точки, видимые из семантической сети через

ссылки от узлов типа «документ». В общем случае содержимое не-RDF-документов не влияет на набор отражаемых семантической сетью сущностей и на структуру этой сети.

Базовую архитектуру мы конструируем исходя из предположения, что все используемые информационные единицы (RDF и не-RDF) опубликованы и могут быть прочитаны агентом по заданной координате. В таком подходе есть серьезная уязвимость: документы или их группы могут обладать большими и очень большими объемами, и такая перекачка будет проблематичной. Для компенсации указанной трудности предполагается активное применение механизмов кеширования и использования копий документов вместо оригиналов по описанным ранее принципам. Также предполагается, что отдельные опубликованные RDF-документы имеют относительно небольшие размеры, т.е. при необходимости разбиваются на части. По крайней мере, такое предположение мы использовали при введении понятия директории.

Другим способом борьбы с трафиком является уменьшение объема передаваемой информации за счет избавления от избыточной информации и сжатия данных. Здесь мы этот вопрос больше затрагивать не будем.

В базовой архитектуре мы предполагаем лишь использование данных информационного пространства программами или людьми. Такой подход аналогичен технологии WWW, где основа технологии — это использование ресурсов, а их создание и изменение может осуществляться внешними средствами. Целью базовой архитектуры является для человека предоставление информации в удобном для него виде, для программного агента — возможность построения рабочей семантической сети или запрос групп фактов.

Интерфейс пользователя — это специализированный браузер (RDF-вьюера), позволяющий осуществить поиск информации и представление найденной информации в удобном для него виде. Такой вьюер естественно создать с использованием стандартного WWW-браузера, но также можно иметь и специализированное приложение.

Для программных агентов требования к интерфейсу разнообразнее. Минимальным интерфейсом, достаточным для создания моста между рассматриваемым информационным пространством и агентом, являются спецификации базовой архитектуры, определяющие совокупность ограничений и регламентаций на публикуемые документы, включая особенности RDF-документов. При этом программный агент может пользоваться или не пользоваться менеджерами доступа и сервисами информационных хранилищ, о которых пойдет речь далее. Кроме того, могут быть представлены

для использования программные интерфейсы (API), берущие на себя большее или меньшее множество функций доступа к данным и обработки данных.

Как было показано, при некоторых предположениях на структуру информации и метаинформации, имеется эффективный алгоритм определения координаты экземпляра документа по идентификатору документа. Поэтому разумно ввести в базовую архитектуру функциональную подсистему «менеджер доступа», реализующую такой алгоритм. В силу того что чтение метаинформационных записей сопровождается предварительным чтением соответствующих RDF-документов, менеджер доступа будет, как правило, иметь свой кэш, сохраняющий прочитанные документы. Роль менеджеров доступа в архитектуре системы в чем-то аналогична роли DNS-серверов и Proxy-серверов в традиционных сетевых архитектурах. Таким образом, базовая архитектура может быть представлена следующим образом.

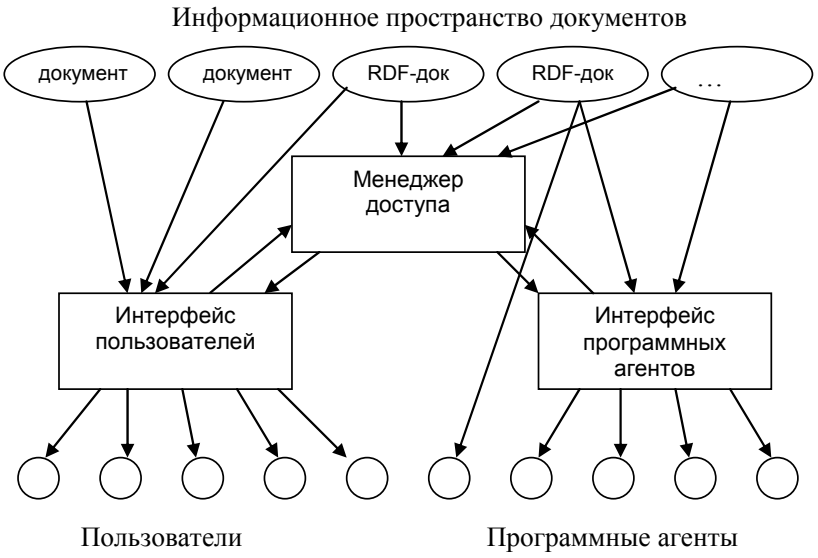

Рис. 3

На рис. 3 изображено взаимодействие пользователей (людей) и программных агентов с информационным пространством документов. При этом, просто документы — это опубликованные файлы, достаточно произвольного вида, RDF-документы — основная система определений онтологий, данных и метаданных. Информация в RDF-документах организована по некоторым зафиксированным принципам (см. приложение 1).

Пользователи получают доступ к информации через интерфейс пользователей, агенты — либо напрямую, либо через интерфейс программных агентов. Оба интерфейса могут пользоваться запросами к менеджеру доступа, который «разбирается» в иерархии виртуальных директорий и адресах экземплярах документов.

МОДЕЛЬ МИРА

Модель мира призвана отражать сущности реального и идеального мира в виде семантической сети, формируемой, например, слиянием RDF-документов. Поскольку речь идет о построении целостной базы данных с фиксацией разнообразных сущностей и установлением прямых связей между сущностями, требуется принципиально иная схема данных, нежели предлагают имеющиеся стандарты на схемы данных DC, GILS, CIMI и др. [8, 9, 10, 11].

В результате было принято решение о формировании онтологии неспецифической информации и базировании разрабатываемых систем на этой онтологии. Путем экспериментов и сопоставлений была предложена методология структуризации данных, основные элементы которой следующие:

- выделение минимального количества ортогональных сущностей, в совокупности характеризующих наиболее существенные моменты описываемых явлений;
- разделение, хотя и в достаточной мере условное, определений на определения сущностей и определения отношений между сущностями;
- отказ, для большинства случаев, от множественности одноименных предикатов (семантических дуг), «выходящих» из одного узла, через обратные ссылки, обладающие свойством единственности;
- усложнение используемых отношений и их «симметризация».

Эти методологические принципы требуют комментариев. Во-первых, выяснилось, что детализация модели мира, предназначенной для общих

применений, не может быть слишком глубокой. В какой-то момент появляется неоднозначность, связанная с тематическими, смысловыми и даже культурными различиями. Например, в понятие «семья» вкладывается несколько отличный смысл в разных ситуациях и разных странах. Отражение таких нюансов часто возможно, но может лишить формируемые данные удобств в задании критериев выборки и возможностей фильтрации выбираемых единиц.

Следующий момент связан с сущностями и отношениями. Базово, в модели Semantic Web, сущности — это узлы, отношения — дуги в графе. Практические прикидки показывают, что важные для описания ситуации отношения, как правило, имеют свою структуру. Механизм reification, зафиксированный в RDF для этих целей, не показался удобными, поэтому базовые отношения, например «персона—организация», получили свои типы и структуру.

Пока не до конца проработанным остается вопрос с «арностью» исходящих дуг. Представляется, что сведение структуры графа к структуре, обладающей свойством единственности (или отсутствия) исходящих дуг одного типа, имеет свои преимущества при построении понятных пользователю интерфейсов.

Теперь об усложнении используемых отношений. Здесь есть два аспекта. Первый — отношения часто, даже скорее как правило, «ведут себя» как сущности. Например, они могут иметь поля или свойства, они могут быть локализованы во времени, они могут соотноситься с другими сущностями, их можно отражать в документах, о них можно делать высказывания. Например, отношение между персоной-работником и организацией-работодателем, как правило, имеет ограниченные временные рамки. В данном отношении может присутствовать такое поле, как «должность», с данным отношением могут быть сопоставлены некоторые документы, например приказы. Второй аспект связан с «бедностью» базового механизма RDF в указании отношения «субъект—предикат—объект». Реальные задачи структуризации требуют не только такого бинарного предиката, но и отношений, построенных на унарных и n-арных предикатах, причем, как уже отмечалось ранее, содержащих дополнительные поля.

OWL-спецификация используемой модели мира приводится в приложении 1. Кратко охарактеризуем основные черты предлагаемой модели. Модель является иерархической по системе определяемых классов. Корневой сущностью является класс Entity. Далее определяется в некотором смысле минимальное множество базовых сущностей, которыми являются:

- персона,
- документ,
- геосистема,
- организационная система,
- коллекция.

Некоторые естественные специализации, такие как страна, город, регион (для класса «геосистема»), организация (для класса «организационная система»), порождают иерархию классов и фиксируют сложившиеся представления о различных частных случаев.

Базовая система отношений включает в себя:

- именование,
- титул (для персоны),
- авторство,
- датирование,
- размещение,
- коммуникация,
- отражение,
- элемент коллекции,
- обучение,
- работу.

Конечно, базовая система сущностей и отношений охватывает только наиболее общие представления о внешнем мире, и создание информационных систем даже общего назначения требует прагматических конкретизаций. Например, под (унарным) отношением «коммуникация» понимаются: телефон, e-mail, почта и т.д. Еще более существенны расширения при предметной ориентации информационных систем. Тем не менее, при таком подходе основные зафиксированные в информационном поле факты могут быть использованы «чужими» информационными системами в соответствии со смыслом этих данных.

ВЫВОДЫ

В данной работе сделана попытка обрисовать необходимость и возможность построения информационных систем нового поколения, основанных на подходе Semantic Web. Для ряда задач, например, электронных архив-

ных и музейных систем, такой переход является необходимостью. Для более «динамичных» задач, например задач поддержки делопроизводства, возможности и предложения, рассмотренные в данной работе, являются проблематичными, хотя само направление модернизации технологий в сторону формирования единого пространства фактов также актуально.

Подход был опробован на ряде проектов. В частности, на его основе созданы:

- простая серверная исполняющая система (движок) по работе с RDF-данными, спецификациями (онтологиями) данных;
- экспериментальный универсальный редактор, настраивающийся на предметную область через заданную онтологию;
- информационная система поддержки работы кафедры;
- система исторической фактографии с наполнением данными по истории ИСИ.

Подход показал свою перспективность и развивается в новых проектах.

СПИСОК ЛИТЕРАТУРЫ

1. **Berners-Lee T., Hendler J., Lassila O.** The Semantic Web // Scientific American. — 2001. — Vol. 284(5). — P. 34–43.
2. **Semantic** Web. — `http://www.w3.org/2001/sw`
3. **Resource** Description Framework (RDF). — `http://www.w3.org/RDF`
4. **RDF Vocabulary** Description Language 1.0: RDF Schema. — `http://www.w3.org/TR/rdf-schema`
5. **Web** Ontology Language (OWL). — `http://www.w3.org/2004/OWL`
6. **Марчук А.Г., Осипов А.Е.** К вопросу об идентификации электронных документов и коллекций // Программирование. — 2000. — № 3. — С. 53–62.
7. **Naming** and Addressing: URIs, URLs. — `http://www.w3.org/Addressing`
8. **Weibel S. L., Lagoze C.** An element set to support resource discovery. The state of Dublin Core: January 1997 // Intern. J. on Digital Libraries. — 1997. — Vol. 1, № 2. — P. 176–186.
9. **Application** Profile for the Government Information Locator Service (GILS), Ver. 2, Nov. 24, 1997. — (`http://www.gils.net/prof_v2.html`)
10. **The CIMI Profile.** Release 1.0H. A Z39.50 Profile for Cultural Heritage Information (`http://www.cimi.org/public_docs/HarmonizedProfile/HarmonProfile1.htm`)/

Приложение 1

СПЕЦИФИКАЦИИ СИСТЕМЫ И БАЗОВАЯ ОНТОЛОГИЯ

Файлы, имена, публикации

1. Любой файл может быть опубликован в информационном пространстве как документ. В этом случае публикатор берет на себя дополнительную заботу: породить об этом файле метаинформационные записи по зафиксированным правилам и поддерживать публикационную корректность как самого файла, так и метаинформации о нем.

2. Кроме опубликованных произвольных документов, информационное пространство содержит множество опубликованных RDF-документов специально определенной конструкции. Это множество и порождает структурирование информационного пространства.

3. Предоставителем информации назовем участника единого информационного пространства, публикующего свои RDF-документы и зарегистрированного в каталогах системы.

3. Имена (идентификаторы) сущностей назначаются предоставителями информации. Правила формирования имен следующие: идентификатор является начинающейся с имени Интернет-домена цепочкой локальных идентификаторов, разделенных символом “/”.

4. Сущности являются типизированными, т.е. для каждого идентификатора сущности имеется определитель rdf:type, указывающий на типовое определение. Не допускается множественного определения типа сущности, даже если ссылки rdf:type совпадают.

5. Для каждой сущности имеется один и только один «владелец» определения. Владельцем является RDF-документ, в котором имеется определяющий rdf:type.

6. Каждый RDF-документ идентифицирован по схеме:
<rdf:RDF iis:ID=`полный идентификатор данного RDF-документа` …>…</rdf:RDF>

7. Полный идентификатор RDF-документа формируется из имени RDF-домена и локального идентификатора документа, разделенных символами «/../». Например, пусть есть домен:

…/alpha/beta

и домен содержит RDF-документ с локальным именем 914. Тогда этот документ будет иметь структуру:

<rdf:RDF iis:ID=`…/alpha/beta/../914`> … </rdf:RDF>

при этом, идентификатор …/alpha/beta будет идентифицировать домен, а идентификатор …/alpha/914 будет идентифицировать RDF-документ, входящий в данный домен.

8. Определения сущностей, имеющиеся в RDF-документе, должны содержать идентификатор, состоящий из имени домена и локального имени данной сущности в домене, разделенные символом `/`. Такое соглашение дает возможность по идентификатору сущности всегда найти хотя бы ее определяющее вхождение.

9. Закрепим за именами, определяющими данную систему соглашений, пространство имен:

xmlns:iis=’http://iis.nsk.su/ns/’

10. Членами директории (как множества) могут быть только RDF-документы, причем RDF-документ должен быть членом одной и только одной директории.

11. У RDF-документа должен существовать хотя бы один экземпляр и точно один оригинал. Оригинал вместе с базовыми свойствами должен располагаться в содержимом экземпляров (файлов) непосредственно вышестоящей директории. По координате, имеющейся в оригинале RDF-документа, должен располагаться файл, содержащий корректный RDF-документ указанной выше конструкции.

12. В записи, касающейся экземпляра документа, должна быть одна запись iis:from-date, указывающая время последней модификации документа.

Схема данных

Базовыми классами и отношениями (properties) данной системы организации информационного пространства являются:

```
<rdfs:Class rdf:about=’http://iis.nsk.su/ns/Any’/>
```

— суперкласс для всех классов сущностей, имеющий следующие базовые отношения;

```
<rdfs:Property rdf:about=’http://iis.nsk.su/ns/name’>
    <rdfs:domain rdf:resource=’http://iis.nsk.su/ns/Any’/>
    <rdfs:range rdf:resource=’http://iis.nsk.su/ns/text’/>
</rdfs:Property>
```

— отношение именования;

```
<rdfs:Property rdf:about='http://iis.nsk.su/ns/from-date'>
    <rdfs:domain rdf:resource='http://iis.nsk.su/ns/Any'/>
    <rdfs:range rdf:resource='http://iis.nsk.su/ns/date'/>
</rdfs:Property>
<rdfs:Property rdf:about='http://iis.nsk.su/ns/to-date'>
    <rdfs:domain rdf:resource='http://iis.nsk.su/ns/Any'/>
    <rdfs:range rdf:resource='http://iis.nsk.su/ns/date'/>
</rdfs:Property>
```

— отношения, задающие начальную дату сущности и конечную дату;

```
<rdfs:Property rdf:about='http://iis.nsk.su/ns/description'>
    <rdfs:domain rdf:resource='http://iis.nsk.su/ns/Any'/>
    <rdfs:range rdf:resource='http://iis.nsk.su/ns/text'/>
</rdfs:Property>
<rdfs:Property rdf:about='http://iis.nsk.su/ns/comment'>
    <rdfs:domain rdf:resource='http://iis.nsk.su/ns/Any'/>
    <rdfs:range rdf:resource='http://iis.nsk.su/ns/text'/>
</rdfs:Property>
```

— отношения, задающие текстовое описание сущности и произвольный текстовый комментарий;

```
<rdfs:Class rdf:about='http://iis.nsk.su/ns/Document'>
    <rdfs:subClassOf
rdf:resource='http://iis.nsk.su/ns/Any'/>
</rdfs:Class>
```

— определяет базовую сущность «документ»;

```
<rdfs:Class rdf:about='http://iis.nsk.su/ns/RDF-document'>
    <rdfs:subClassOf
rdf:resource='http://iis.nsk.su/ns/Document'/>
</rdfs:Class>
```

— частный вариант документа — RDF-документ;

```
<rdfs:Class rdf:about='http://iis.nsk.su/ns/Set'>
    <rdfs:subClassOf
rdf:resource='http://iis.nsk.su/ns/Any'/>
</rdfs:Class>
```

— множество произвольных сущностей;

```
<rdfs:Property rdf:about='http://iis.nsk.su/ns/member-of'>
    <rdfs:domain rdf:resource='http://iis.nsk.su/ns/Any'/>
    <rdfs:range rdf:resource='http://iis.nsk.su/ns/Set'/>
</rdfs:Property>
```

— множество задается элементами, которые «связываются» с ресурсом ;

```
<rdfs:Class rdf:about='http://iis.nsk.su/ns/Directory'>
    <rdfs:subClassOf
rdf:resource='http://iis.nsk.su/ns/Set'/>
</rdfs:Class>
```

— директория — специфическое множество, группирующее RDF-документы;

```
<rdfs:Class rdf:about='http://iis.nsk.su/ns/Document-
example'>
    <rdfs:subClassOf
rdf:resource='http://iis.nsk.su/ns/Any'/>
</rdfs:Class>
```

— экземпляр документа — место («файл») для «хранения» содержимого документа;

```
<rdfs:Property rdf:about='http://iis.nsk.su/ns/example-of'>
    <rdfs:domain
rdf:resource='http://iis.nsk.su/ns/Document-example'/>
    <rdfs:range
rdf:resource='http://iis.nsk.su/ns/Document'/>
</rdf:Property>
```

— отношение, задающее связь между экземпляром документа и документом;

```
<rdfs:Property rdf:about='http://iis.nsk.su/ns/last-example-is'>
    <rdfs:domain
rdf:resource='http://iis.nsk.su/ns/Document'/>
    <rdfs:range rdf:resource='http://iis.nsk.su/ns/Document-
example'/>
</rdfs:Property>
```

— указание на то, какой экземпляр документа является оригиналом («последним» экземпляром);

```
<rdfs:Property rdf:about='http://iis.nsk.su/ns/coordinate'>
    <rdfs:domain
rdf:resource='http://iis.nsk.su/ns/Document-example'/>
    <rdfs:range rdf:resource='http://iis.nsk.su/ns/uri'/>
</rdf:Property>
```

— координата экземпляра документа.

Реальные онтологии неспецифической и специфической информации, использованные в экспериментах, отличаются от приведенной. Автор также далек от мысли, что рассмотренная схема данных (онтология) будет без изменения использоваться в дальнейшем. Рассмотренный фрагмент базовой онтологии приведен в качестве иллюстрации подхода, в настоящее время использованного для формирования схем данных, подхода, который уже подтвердил свою эффективность на ряде проведенных экспериментов и созданных информационных систем.

А.Г. Марчук

РАСПРЕДЕЛЕННЫЕ ЭЛЕКТРОННЫЕ АРХИВЫ, БИБЛИОТЕКИ И БАЗЫ ДАННЫХ

Препринт
122

Рукопись поступила в редакцию 27.11.04
Редактор З. В. Скок

Подписано в печать 25.12.04
Формат бумаги 60 × 84 1/16 Объем 1.5 уч.-изд.л., 1.6 п.л.
Тираж 60 экз.

ЗАО РИЦ «Прайс-курьер»
630090, г. Новосибирск, пр. Акад. Лаврентьева, 6, тел. (383) 330-72-02