О распределенных фактографических системах

© А.Г. Марчук

Институт систем информатики им. А.П. Ершова СО РАН, НГУ
mag@iis.nsk.su

Аннотация

В работе рассматриваются вопросы, связанные с фиксацией и структуризацией фактов. Речь идет о модели, создаваемой в виде набора сущностей вида: персоны, организационные системы, географические системы, документы и некоторые другие, записи существенных атрибутов этих систем и установления отношений между сущностями. Подход ориентирован на процессы, развивающиеся во времени и пространстве и позволяет создавать базы данных не устаревающие со временем и не зависящие от места восприятия и позиции воспринимающего.

Фактографический подход применим к ряду прикладных задач таких, как создание электронных архивов, музеев, исторических энциклопедий и справочников. Также была показана его эффективность к созданию простых систем делопроизводства. Изучается возможность использования структурированного поля фактов для создания аналитических систем и систем извлечения знаний.

Особенностями подхода являются: использование концепции Semantic Web [1], создание онтологии неспецифических сущностей, ориентация на распределенный характер создаваемого информационного поля, применение новой концепции информационного пространства документов [2].

Предлагаемые подходы и решения были частично реализованы в ряде прикладных проектов, выполненных и выполняемых в Институте систем информатики СО РАН.

1 Введение

Законченные полные версии принятых статей необходимо подать в оргкомитет до 14 июля 2008 года, используя форму подачи финальной.

Одно из самых дорогостоящих в современной информационной индустрии – это данные, собираемые и накапливаемые для решения конкретных задач. Проблема заключается в том, что подавляющее большинство источников данных плохо сочетается между собой и существует огромное дублирование информационных единиц, описывающих одно и то же. Например, все мы, как абоненты разных сервисов и порталов Интернета зарегистрированы в разных базах данных своей персональной информацией, тем не менее, очередная заинтересовавшая нас информационная система снова предлагает ввести данные очень похожего профиля. Нам говорят, что персональные данные являются приватными и существует законодательство об ограничении распространения персональной информации. Все правильно, но та же ситуация сохраняется и с теми данными, которые не подпадают под ограничения, являются по существу публичными и, на самом деле, предназначены для распространения. Это относится к данным об организациях, о географических объектах, об открытых документах. Легко сообразить насколько интереснее, актуальнее и, соответственно дороже были бы данные, размещенные в едином источнике, заслуживающем доверия. Назовем эту проблему проблемой формирования единого информационного поля (фактов).

Некоторые достижения в направлении формирования единого информационного поля имеются. Для регистрации информационных сервисов имеется система UDDI [3], для унификации новостного потока используется RSS, для описания информационных ресурсов широкого профиля – Dublin Core [4], для частных случаев таких ресурсов, есть стандарты типа MARC, CIMI [5] и др. Проблема видится в отсутствии целостного подхода к структуризации разносортных данных, который позволил бы человеку или машинному агенту надежно находить требуемую ему информацию и использовать для решения тех или иных задач.

Другим аспектом группы проблем, находящихся в рассмотрении в данной работе является правильность устройства данных. Изучение применяемых схем данных показывает, что во многих случаях их разработка велась без опоры на какие-либо принципы кроме принципов, связанных с нормализацией и других очевидных аспектов. Наиболее распространенной ошибкой разработчиков базы данных является видение ситуации как мгновенного среза. Например, в базе данных сотрудников организации зафиксированы только нынешние сотрудники с указанием даты приема в организацию. А как же бывшие сотрудники? База данных сразу бы приобрела

Труды 10-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» – RCDL'2008, Дубна, Россия, 2008.

черты исторической. Вроде такую модификацию проделать довольно просто, достаточно ввести поле завершения пребывания в организации. Однако, все не так просто. Довольно быстро выявляется группа новых проблем, препятствующих введению в базу данных историчности. Оказывается, работник не пребывал в одной должности и она как-то менялась, некоторые работники (девушки), меняли свои фамилии выходя замуж, оказывается нестабильно все – со временем менялись паспортные данные, телефоны, места проживания, научные степени, наконец, – структура предприятия, название предприятия.

Предлагаемый подход мы называем фактографическим, т.е. подходом, основанным на записи фактов. Заметим – на записи, а не стирании или редактировании. Мы ставим перед собой задачу сформировать принципы структуризации разносортных данных, при которых все ранее записанные факты сохраняют свою справедливость вне зависимости от времени, географии или других факторов воспроизведения этих фактов. А изменение базы фактов осуществляется лишь добавлением новых информационных единиц (фактов). Простой пример: традиционное указание в поле записи о персоне возраста как числа лет – неправильный способ фиксации данных, правильный – указание даты рождения. К этому полю, в дальнейшем, будет добавлено другое поле: даты завершения жизненного пути. Поставленная цель идеальна. Реально в реализации имеются средства и изменения информационных полей и связей и их уничтожения. Однако, как будет показано далее, эти традиционные редактирующие действия приобретают новый смысл.

Наиболее естественным данный подход является в задачах исторического профиля, когда фиксируются факты, относящиеся к большому временному периоду и часто – к большой географической разбросанности. Принципы структуризации и основные методические моменты были разработаны при формировании и ведении базы данных кафедры программирования НГУ, где динамика изменений – очень большая и переход студентов из одной категории в другую исчисляется единицами лет. Данная задача относится к классу делопроизводственных, но ее особенности высветили ключевые проблемы фактографического подхода. В дальнейшем, подход получил реализацию и развитие в ряде проектов исторической направленности, но и тут, сопряжение исторического материала с фактами сегодняшнего дня (делопроизводством) является актуальной проблемой, если исходить из простого соображения, что сегодняшние события завтра станут историей.

Возвращаясь к проблемам приватности информации, в предлагаемом подходе эта проблема только обостряется. Действительно, при достаточно полном наборе информации о разных аспектах персон какого-то социума, сведение воедино разных информационных единиц выглядит пугающим при ее попадании в руки злоумышленников. Например, о персоне могут быть изложены и дата рождения и места работы, проживания, коммуникационные адреса, отношение родства с другими персонами, посещаемые мероприятия, планы этих мероприятий, имена коллег и их телефоны у которых можно получить дополнительную информацию и т.д. и т.п. Причем уверяю вас, по публичным людям практически всю такую информацию можно получить в открытых источниках, потенциальной опасностью обладает именно информация, сведенная воедино. Вот почему наш коллектив предпочитает заниматься делами уже ушедших дней и увы, ушедших людей.

Еще одной особенностью представленного в данной работе подхода является ориентация на создание распределенных систем. Дело в том, что множество информационных ресурсов создается и поддерживается самостоятельными командами, имеющими независимые цели и предпочтения. Как правило, они хотят распоряжаться результатами своего труда в соответствии с собственным разумением, не ориентируясь на внешние процессы. Самый простой вариант для этого случая – полностью автономная разработка, что имеет существенные минусы: во-первых, не всегда используются профессиональные реализационные решения, во-вторых, созданный информационный массив не пополняет общего информационного поля, в-третьих, судьба информационного ресурса по истечению проекта часто плачевна. Наш подход позволяет решить все три проблемы не меняя автономного характера работы. Подход предлагает ряд методических и технологических решений, которые могут быть выполнены независимо и профессионально, а в дальнейшем развиваться в соответствии с изменениями базовых систем программирования. В подходе предусмотрено разделение информационного поля на приватное и общедоступное, имеются механизмы расширения схемы данных до уровня учета специфики локальной задачи, имеются средства настройки базовых решений на предметную область и предпочтения пользователей. В случае завершения проекта, созданная база данных может быть «втянута» в централизованное хранилище и продолжать участвовать в формировании общего информационного пространства.

2 Единое информационное пространство

В настоящее время существует множество подходов к фиксации фактов исторической направленности. К таки подходам относятся: базы данных, всемирная паутина World Wide Web, системы описания информационных ресурсов, основанные на метаинформации, электронные библиотеки. В последние годы активно развивается новое направление в информационных технологиях: Semantic Web (SW) [1].

Недостатками использования реляционных баз данных для решения задач фиксации фактов являются централизованный характер хранения и обработки, трудности в сопровождении и модернизации,

неудобства работы со спецификациями и словарями. Другая технология – WWW вообще мало ориентирована на формальные подходы к структуризации данных и, в основном, предназначена для использования информации, выполнения поиска и навигации человеком. Существенное развитие произведено в системах структуризации, основанных на метаинформации и в электронных библиотеках [6]. Проблемой этого подхода является «плоский» характер структуризации, больше предназначенный для больших объемов однородной информации и, как правило, не порождающий базы данных разносортных элементов. В отличие от упомянутых подходов, методология SW дает возможность формализовать существенные знания о смысле данных и использовать эти знания для создания информационных систем нового поколения, позволяет создавать распределенные системы, основанные на полномасштабной базе данных. Поле данных, при этом, легко использовать как для формирования WWW-интерфейсов, так и для выполнения запросов от машинных агентов.

В качестве базового подхода для широкого класса исторических фактографических систем был определен подход Semantic Web. Это обусловлено тем, что в подходе комплексно могут быть решены проблемы использования несвязанных или слабосвязанных между собой источников информации. Кроме того, формализмы, имеющиеся в SW и прилегающих стандартах, адекватны задаче работы с распределенным полем разносортной информации. Работа с данными, специфицированными средствами OWL [7], отличается также возможностью выполнения логического вывода для ряда практически важных случаев, а также применения других технологий, относящихся к искусственному интеллекту. В настоящее время ведутся исследования по использованию программирования в ограничениях, нечеткой логики, семантическому выводу и кластеризации данных.

Предлагаемый подход подразумевает формирование модели, в которой сущности внешнего мира представляются атрибутированными информационными единицами, отношения между сущностями реализуются либо в виде прямых ссылок, либо в виде составных конструкций определенного вида. Спецификация такой модели, воплощенная в виде онтологии позволяет создавать более универсальные программы обработки, использовать спецификации для удобного представления данных, многоязыковой реализации информационных систем, для анализа данных на предмет полноты и корректности. Важным элементом технологии RDF, которая находится «в центре» методологии SW, является концепция формирования семантической сети из отдельных высказываний и групп высказываний. Объединение высказываний в единый граф выполняется на основе слияния одинаково идентифицированных информационных единиц (айтемов). Это дает легкую возможность объединять воедино RDF-модели, хранимые в разных местах, то есть, порождает естественную распределенность информационного поля.

Рассмотрим особенности построения распределенного информационного поля. С точностью до деталей, информационное поле можно представлять как набор RDF-документов [8], логически связанных между собой единой идентификацией одинаково понимаемых объектов и отношений. Будем использовать термин «публикация» для характеризации того, что документ расположен по фиксированному для него URL и «виден» из Интернета. Возможны детали, связанные с возможным ограничением видимости для разных категорий пользователей, а также возможным косвенным характером доступа к документу. Последнее означает, что допустима ситуация, когда например реляционная база данных, погруженная в СУБД, имеет специальный интерфейс, преобразующий базу данных или ее часть в запрашиваемый RDF-документ. Опубликованные документы противопоставляются неопубликованным, т.е. таким, доступ к которым возможен только для «своих».

Таким образом, любой интернетовский агент имеет равный доступ к общим опубликованным документам и локальный доступ к своим документам. Из этого множества регулярно построенных документов агент может сформировать базу данных в виде (семантической) сети или, в некоторых случаях, в виде системы реляционных таблиц. Использование такой собранной базы данных может происходить в соответствии с функциями того агента, который собрал базу данных. В итоге, построена общая схема организации сосуществования информационных систем, обладающих собственным информационным контентом, при этом опирающихся на общее информационное пространство.

Общее пространство фактов будет полезным только если будет решена проблема «понимания», т.е. если каждый из документов с общими данными будет использовать структуризацию данных, сводимую к той, которая используется конкретным агентом (конкретной информационной системой). Но это еще не все. Нужно также, чтобы объекты разных баз данных, означающие одинаковые сущности, смогли бы объединиться при слиянии баз данных. Первую задачу назовем задачей совместимости онтологий, вторую задачу – проблемой отождествления.

В самом «жестком» варианте, единое информационное пространство выстраивается из RDF-документов, структурированных в соответствии с единой онтологией, а отождествление выполняется базовым для RDF способом – через одинаковые идентификаторы сущностей. В «мягком» случае, данные из подгружаемых документов подвергаются переструктуризации в соответствии с различиями в онтологиях, а проблема отождествления решается каким-то алгоритмическим способом. Заметим, что при недостаточности информации, алгоритм отождествления вообще может не дать положительного решения для слишком большого количества случа-

ев. Действительно, мировая практика идентификации персон предполагает указание для конкретной персоны полного имени, места рождения и даты рождения, последние две позиции, относительно конкретной персоны, вообще могут отсутствовать в базе данных. Не забудем также, что имя может писаться на разных языках, что имя может измениться и т.д. и т.п.

Можно сделать вывод, что «мягкий» подход к обеспечению совместимости данных может оказаться непрактичным. Наш подход в решении задачи совместимости онтологий заключается в том, что для разных документов общего информационного пространства допускается использование разных онтологий, но лишь в виде расширения общего онтологического построения, которое мы называем базовой онтологией. Здесь уместно привести аналогию из области использования библиотек компонентов для создания специализированных программных комплексов. Если этих библиотек много и они не согласованы между собой, то их использование довольно затруднительно. Если же библиотеки выстраиваются в иерархию по принципу послойного расширения, то при независимости (ортогональности) таких расширений, множество специальных программных комплексов могут эффективно их использовать для достижения своих целей.

Проблему отождествления мы решаем следующим образом. Предпринимаются меры для того, чтобы отождествление производилось на наиболее ранних этапах формирования конкретных доменов данных, например при первичном вводе данных, особенно если ввод осуществляется вручную. Этому могут помочь соответствующие программные средства и «тотальные» базы данных типовых сущностей, таких как персоны, организации, города и др. Тем не менее, для реальных ситуаций, когда данные вводятся независимыми процессами, всегда остаются случаи, требующие алгоритмического решения задачи отождествления, соответствующие инструменты должны присутствовать в арсенале систем редактирования и манипуляций.

Возникает вопрос: а правильно ли формировать целостную модель (базу данных) в каждом из мест обработки? Нет ли возможности распределить не только данные, но и обработку этих данных? Представляется, что некоторые подходы к этой задаче возможны, однако в целом, задача является математически и методически открытой и, в данной работе, мы ее обсуждать не будем.

3 Базовая онтология

Ключевым вопросом в реализации подхода, явился вопрос создания онтологии неспецифических (базовых) сущностей. Термин «неспецифических» используется в смысле противопоставления «специфическим» сущностям. Дело в том, что практически каждая информационная система обладает своим предметом, своей спецификой. При этом, наряду со специфическими сущностями, например какими-нибудь номерами счетов, транзакциями и остатками средств, очень часто в онтологии необходимо также иметь такие классы сущностей, как персоны, организации, места нахождения, коммуникационные координаты, т.е. сущностей, являющихся продуктом общекультурного взгляда на мир и современную цивилизацию, лишенные конкретной специфики и одинаково понимаемыми создателями систем и пользователями.

К сожалению, на данный момент, подобной онтологии неспецифических сущностей как мирового стандарта или рекомендации – не существует, хотя в ряде стандартов (DC, FOAF [9], CIMI) прослеживаются ее элементы. Исходя из сформулированных далее требований и принципов, нами была порождена и обоснована базовая онтология, которая в настоящее время используется в ряде проектов.

Базовая онтология формировалась в течение довольно длительного времени при работе над реальными проектами. И можно точно утверждать, что ее построение еще не закончено. Накапливавшиеся в процессе эволюции онтологии изменения требовали периодической трансформации накопленных данных, но методы абстрактного программирования, основанные на использовании явных спецификаций данных, позволяют сохранить при таких изменениях большую часть кода программ и интерфейсов.

Был выработан набор принципов построения онтологий, в частности, базовой онтологии.

1. **Отражение фактов реального мира.** При создании общего информационного поля лучше придерживаться фактографического способа формирования модели мира. Это означает, что предметом фиксации в базе данных должны быть реально существующие сущности, причем сведения об этих сущностях должны иметь объективный характер, не зависящий от особенностей предметных областей, для которых эти факты используются. Например, если мы создаем историческую базу данных, то при отражении конкретной персоны, нас интересуют такие факты, как дата рождения, полученное образование, места работы и проживания. Но мы воздержимся от фиксации таких субъективных моментов как оценка достижений человека, его популярности или влиятельности. Эти моменты могут возникнуть при оформлении конкретного взгляда на исторические процессы, фиксируемого конкретной информационной системой, которая, как уже отмечалось, может активно использовать общее информационное поле фактов.

2. **ER-модель как основа онтологического построения.** Традиционная модель Entity – Relationship является и естественной и эффективной при данном подходе. Действительно, отражая сущности и атрибуты сущностей, мы порождаем поле независимых информационных единиц. Отражая отношения между сущностями, мы формируем связанную базу данных, в которой от одного объекта можно «добраться» до другого, что соответствует устройству мира. Если база данных распадается на

независимые части, это свидетельствует либо о неполноте данных, либо о неправильной структуризации данных.

3. **Балансировка по детальности описаний**, минимизация национальных и культурных различий. Описать мир во всех подробностях невозможно и нецелесообразно. Поэтому надо ограничить детальность описаний некоторой глубиной. Это относится и к набору сущностей и к набору отношений. Детальность описаний также лимитируется возможностями одинакового понимания вводимых понятий и градаций разными группами разработчиков и пользователей. Например, виды организационных структур могут весьма различаться в разных странах и даже в одной стране в исторической ретроспективе. Также, руководящие должности, территориальное административное деление, виды документов, родственные отношения и т.д., могут называться очень по-разному и не будут поняты или обработаны при анализе данных разный временных периодов и стран.

4. **Специальная реализация атрибутированных отношений.** Одной из распространенных ошибок, совершаемых создателями онтологий или схем баз данных является игнорирование возможных атрибутов для конкретного отношения. Для онтологий, создаваемых средствами OWL, такое ошибочное решение обычно воплощается в использовании в качестве отношения прямой ссылки через определение объектного свойства. Например, часто отношение «работа» между персоной и организацией рассматривается как объектное свойство. Это неправильно. Человек вступает в отношение с организацией в конкретный момент времени, может расторгнуть эти отношения в другой, в отношении «работа» могут быть и другие атрибуты, например должность.

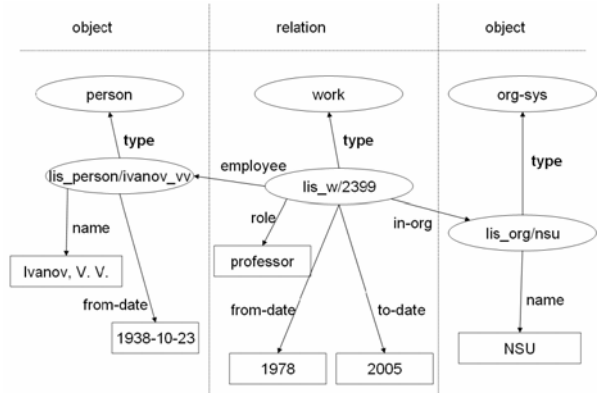

Рис. 1. Пример связывания объектов атрибутированным отношением

5. **Прямая фиксация контекстной информации**. Люди привыкли излагать свою мысль в определенном контексте. Например, «Путин сказал…». Совершенно очевидно, что имеется в виду вполне конкретный человек из десятков тысяч Путиных, имеется в виду недавнее время и, скорее всего, вне этого предложения обрисовано мероприятие, место его проведения и другие контекстные особенности. В слишком большом количестве случаев некоторый контекст остается и в используемых формальных спецификациях данных. Это резко сужает возможность объединения данных, созданных в разных неявно присутствовавших контекстах. Часто делается попытка использовать контекстную информацию для группирования информационных единиц, например, объединение в списки персон, работающих или проживающих в одном месте, группирование по ролям (преподаватели-студенты, врачи-больные, работники, сгруппированные по иерархии организационной структуры предприятия и др.). Анализ показывает, что использование подобных группирующих или иерархических отношений в качестве прямой основы структуризации данных никакой полезности не несет, но может существенно усложнять доступ к данным. Подобные группирования являются вторичными и должны выводиться из записанных в базе данных свойств данных и отношений между ними.

6. **Сводимость базовой онтологии к системе простых реляционных таблиц**. Этот принцип означает, что онтология формируется так, чтобы средой структуризации и хранения данных могли бы быть не только RDF-документы, но и реляционные таблицы простого вида, причем соответствие между ними было бы эффективным в смысле реализации. Этот принцип видится важным для обеспечения совместимости формируемых RDF-данных с существующими реляционными данными и для использования реляционных СУБД как среды для поддержания модели семантической сети. Такое соответствие видится достаточно естественным: каждому классу сущностей сопоставляется таблица, первичным ключом которой является идентификатор сущности, используемые поля данных (атрибуты, DatatypeProperty) превращаются в типизированные колонки, прямые ссылки (ObjectProperty) превращаются в колонки внешних ключей. Такое соответствие существенно ограничивает набор принимаемых решений при формировании базовой онтологии. Во-первых, приходится отказаться от множественности для указания типа сущности, кроме того, онтология требует обязательной типизации сущностей. Во-вторых, не используется наследование для свойств DatatypeProperty и ObjectProperty. В-третьих, на уровне онтологии не используются «сложные» конструкции RDF такие как blank nodes, контейнеры (containers), списки (list), reification. В-четвертых, желательно не допускать множественности значений в колонках реляционных таблиц. Это, как правило, можно сделать, оставляя только «правильные» отношения и управляя направлением стрелок в свойствах, связывающих разные сущности.

В основе созданной базовой онтологии определены четыре класса сущностей: персоны, организационные системы, документы и географические системы. Между базовыми сущностями допустим ряд отношений, общая схема основных отношений изображена на рисунке.

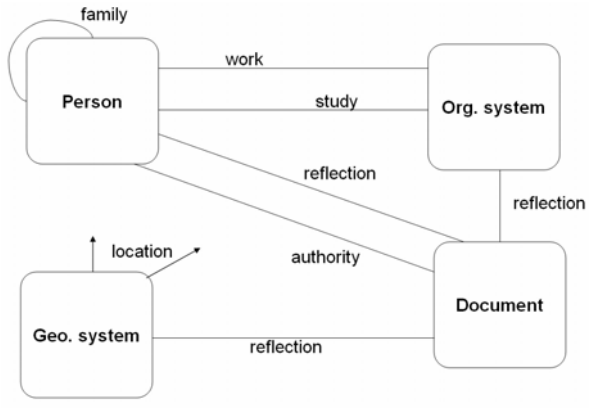

Рис. 2. Основная часть базовой онтологии

Под организационными системами мы понимаем (временное) сообщество людей, ориентированных на достижение конкретных целей. В эту категорию попадают и организации и мероприятия и команды и ассоциации людей и организаций. Под понятие географических систем попадают города, страны и прочие места проявления активности людей. Документы – широкий набор целостных образований (книги, статьи, файлы, фотографии и др.), сущностью которых является зафиксированное информационное наполнение. Между введенными сущностями онтологией определены следующие отношения. Между людьми учитываются только семейные отношения, между персонами и организационными системами возможно отношение «участник», причем в силу важности, выделено специальное отношение «обучение». Отметим, что уже стало традицией, в Curricula Vitae помещать информацию о полученном образовании. Между документами и сущностями других классов, основным отношением является отношение отражения. Под этим мы понимаем, что в содержимом (content) документа изображена или упомянута данная сущность. Кроме того, персоны могут выступать в качестве авторов документа, что фиксируется отношением «авторство». Географические системы выступают в качестве места для какого-то зафиксированного в базе данных момента в виде отношения «размещение» (location).

Из других классов сущностей, имеющихся в базовой онтологии, следует упомянуть коллекции и архивы. Кроме того, на рисунке на изображен ряд унарных и бинарных (сложных) отношений, таких как: родственные организации, титулы, степени и награды, коммуникационные адреса и др. В качестве примера, покажем как для персоны указать адрес электронной почты.

```
mail rdf:about="em134983">
  <user rdf:resource="mag" />
  <e-code>mag@iis.nsk.su</e-code>
  <from-date>1990</from-date>
</email>
```

Легко видеть, что введенный способ «прикрепления» адреса электронной почты к персоне, не ограничивает ни количества адресов, ни возможности «закрытия» устаревшего адреса. Последнее легко выполняется добавлением к записи поля `<to-date>YYYY-MM-DD</to-date>`, указывающего время завершения действия отношения email.

В базовой онтологии широко используется введение подклассов для указания частных случаев и даже некоторые классы помечаются как абстрактные (что делается средствами, дополнительными к формализму OWL). Важно отметить, что в силу использования механизма классов для сложных отношений, показалось, что определенный для корневого класса в OWL термин Thing терминологически не соответствует понятию отношения. Например, отношения «участник», «размещение» и др., являющиеся сложными отношениями, вряд ли логично называть «вещью» (thing). В нашей онтологии, корневой класс является абстрактным и назван entity. Для классов основных сущностей, которые уже соответствуют понятию «вещь», мы используем класс sys-obj – системный объект. От этого абстрактного класса наследуются персоны, организационные системы и др.

Существенным нововведением в базовой онтологии является понятия датирования и именования. Сначала о датировании. Вообще, указание временного диапазона определено прямо для коневого класса, как основное свойство системных объектов и отношений через два атрибута (DatatypeProperty) from-date и to-date, означающие временные отметки начала и конца сущности или отношения. Однако оказалось, что этого недостаточно. Дело в том, что во многих случаях практической работы с данными, точные значения этих атрибутов указать затруднительно. Зато можно сделать высказывания типа: «эта сущность (отношение) началась до такой-то даты», «эта сущность еще не закончилась (или уже закончилась)» и др. Также иногда можно указать приблизительную дату, но это не то же самое, что указать дату точно.

Некоторое решение указанных проблем можно найти во введенном унарном отношении датирования (dating). Отдельным датированием для указанной сущности определяется некоторая дата, которая имеет квалификатор и точность задания этой даты. Квалификатор может быть: before-date, from-date, in-date, to-date, after-date, т.е. один из пяти вариантов отметки точки на временной оси, на которой расположен интервал существования данного явления. Например,

```
<dating rdf:about=”da298345”>
  <referred rdf:resource="em134983"/>
  <date>2006</date>
  <d-specificator>to</ d-specificator>
  <d-accuracy>1-0-0</d-accuracy>
</dating>
```

Означает, что мы отметили временную точку 2006, известную с точностью до одного года, как дату окончания сущности, имеющей идентификатор da298345.

Теперь рассмотрим именование. Для системных объектов имеется стандартный атрибут name, означающий имя объекта. Однако, как и в предыдущем случае, этого оказывается недостаточно. Соответст-

венно, введено унарное отношение naming, означающее сопоставление системному объекту (еще одного) имени или синонима. Кроме того, что мы получили возможность множественного обозначения объекта, именование можно атрибутировать, например можно указывать с какого и по какое время существовал такой вариант имени.

4 Ввод и редактирование данных, визуализация информации

Системы, базирующиеся на формальных спецификациях, обладают существенным достоинством высокого уровня абстракции проектирования. Программист рассуждает не в терминах персон, городов, организаций, а в терминах математической модели, отношение которой с реальным миром задается онтологией, которую он также рассматривает как математический объект. Не всегда удается достаточно глубоко выдержать такой уровень абстракции, но в базовых действиях: визуализации, редактировании поиске, навигации – это возможно и желательно. Полученные программные средства становятся универсальными, для их использования достаточно задать онтологию и некоторые другие спецификации. А полученная система способна «разговаривать» с пользователем в терминах заданной спецификациями предметной области и на языке, удобном пользователю.

Рассмотрим RDF-построение как математический объект и некоторые из тех задач, которые пользователь желает решать.

RDF-модель, в своей основе, представляет собою ориентированный граф, узлами которого являются сущности (в терминологии стандарта – subject) и строковые константы. Дугами являются связи «сущность – сущность» и «сущность – константа». Другие свойства модели – типизацию, идентификацию, соответствие формальным спецификациям (онтологии), мы будем в рассуждениях привлекать по необходимости. Назовем айтемом типизированную сущность нашей модели. Если сосредоточить взгляд на конкретном айтеме, то легко увидеть его каноническое представление:

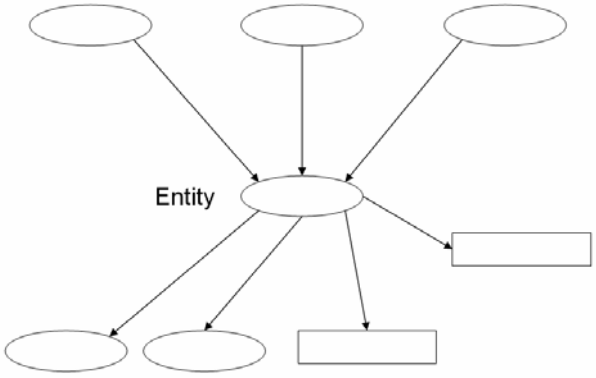

Рис. 3. Каноническое представление айтема

Заметим, что мы ограничиваемся рассмотрением только типизированных моделей, т.е. таких моделей, в которых определены классы сущностей и определены свойства (DatatypeProperty, ObjectProperty) и их возможные использования в модели. Часто мы также ограничиваемся рассмотрением онтологий, в которых множественность (cardinality) исходящих дуг ограничена единицей. Рассмотренная в предыдущем разделе базовая онтология обладает указанными свойствами.

Как мы видим, фокусирование на конкретном айтеме выделяет также подграф, являющийся ближайшим окружением данного объекта. Легко видеть, что в совокупности, атрибуты айтема, вместе окружением айтема, представляют существенную информацию для его характеризации. Назовем такой подграф информационным портретом айтема (сущности). Например, сфокусировав внимание на некоторой персоне, в соответствии с базовой онтологией, мы получим информацию о возрасте, родственниках, образовании, работе, проживании, участии в мероприятиях, коммуникационных адресах и т.д. Иногда, при этом полезно будет «заглядывать» на более, чем одну дугу в окружении изучаемого объекта.

Если ввести в модели метрику, например как взвешенную сумму числа дуг между узлами графа, то задачу построения информационного портрета можно свести к задаче выделения из графа шара определенного диаметра, с центром в айтеме объекта изучения. Такая геометрическая подзадача является не единственной интересной постановкой. Если выделить два узла (айтема), то можно выявлять некоторую окрестность точек, сумма расстояний от которых до фокусов не превышает заданной величины, т.е. строить эллипсоид. Полезная интерпретация данного построения заключается в формировании информационного портрета айтема с точки «зрения» другого айтема типа: что это за объект и что связывает данный объект со мной. Возможны и другие способы определения подграфа, основанные не только на метрике и назначении весовых коэффициентов для дуг, но и учитывающие специально назначенные веса для узлов. Несложно сконструировать оболочку для N назначенных точек и интерпретировать ее как раскрытие заданной темы. Подобные построения будут эффективны как при вычленении содержательной информации при построении справок (досье), так и для организации поисковых действий по множеству ключевых слов или объектов.

Работа с сетевыми структурами RDF имеет свои особенности. Это касается и визуализации информации и поиска и редактирования. Причем все три процесса могут быть перемешаны в рамках единой работы. Начнем работу с поиска, сформулируем поисковый запрос, вид которого нам пока не существенен. По запросу могут быть найдены айтемы-кандидаты, соответствующие критериям запроса. При на слишком полном запросе или при недостаточности информационного покрытия, таких кандидатов может быть несколько. Следующим этапом является визуализация множества айтемов-

кандидатов в таком виде, чтобы пользователь смог найти нужный. Визуализация списка айтемов как правило базируется на визуализации отдельного айтема, вопрос лишь в детальности создаваемого портрета. После выделения найденного айтема, пользователя может интересовать его детальный информационный портрет. Легко видеть из особенностей построения RDF-моделей, что информационный портрет состоит из множества значений атрибутов и множества айтемов ближайшего окружения. Множество значений атрибутов легко реализуется в виде набора пар: имя атрибута – значение атрибута и представимо в традиционной форме «в столбик» или «в строчку (таблицы)». А множество ассоциированных айтемов мы можем визуализировать уже упоминавшимися средствами. К этой композиции достаточно добавить навигационные переходы на айтемы окружения и базовые принципы визуализации семантических сетей RDF, в общем, изложены.

Трудности появляются в процессе редактирования в связи с необходимостью постоянно, на уровне оператора решать две задачи, обе из которых, как минимум, непривычны этому оператору в контексте редакторских действий. Первая – поиск и отождествление информационных единиц, вторая – связывание одних объектов с другими через простые или сложные отношения. В чем состоит непривычность или сложность данных действий для оператора? Дело в том, что на уровне ввода и редактирования данных, наиболее стойкими и привычными конструкциями для ввода и редактирования, являются запись, в случае одиночного объекта редактирования и таблица, для множественности объектов редактирования. В работе с визуальным интерфейсом это означает или редактирование полей формы или редактирование ячеек таблицы. Работа со ссылками – обычно не предусматривается. Значит, проблема заключается в том, как устанавливать именованную связь данного айтема с другим айтемом базы данных. Рассмотрим эту задачу для случая «прямых» ссылок, в графовой модели это соответствует исходящим из айтема дугам, ведущим к другим айтемам. Пусть в ячейке, имеющей ссылочный тип (ObjectProperty), надо средствами действий с визуальным интерфейсом установить конкретную ссылку. Каким должен быть интерфейс ввода ссылки? Причем желательно, чтобы этот интерфейс был как бы «внутри» ячейки и не портил общей портрет редактируемого айтема. Имеется несколько вариантов решения, применяемых в различных системах.

Самый простой способ: превратить набор айтемов-кандидатов на связывание в визуальный элемент интерфейса типа ListBox или ComboBox. Несмотря на то, что этот вариант является самым понятным пользователю, имеется ряд недостатков, сильно ограничивающих его применимость. Во-первых, без введения дополнительной информации со стороны пользователя, список кандидатов на ассоциирование может оказаться непомерно большим, что лишает этот способ практичности. В этом плане, вариант ComboBox предпочтительнее, поскольку в нем встроена возможность ввода поисковой информации. Во-вторых, айтема, с которым мы хотим установить связь может не оказаться в базе данных. Тогда в списке будет фигурировать вариант «другое» и для этого варианта нужна специальная форма инициации ссылаемого айтема. В-третьих, для определения того, какой из айтемов списка является искомым, часто надо посмотреть детали кандидата, что также является нетипичным действием для редактирования.

Другим вариантом является внедрение поискового интерфейса в данное конкретное место редактирования, возможно, прямо в ячейку редактирования. В этом случае, сначала дается форма для формулирования поискового запроса, потом появляется, упоминавшийся ранее, список айтемов из которого требуется сделать выбор и имеется «кнопка» формирования нового айтема, если поиск не выявил его существования. Новый айтем строится исходя из имеющегося контекста и введенной в поисковом запросе информации. Довольно громоздко может выглядеть такое универсальное решение в случае, когда тип прикрепляемого айтема может различаться, тогда, как правило, сначала пользователь устанавливает тип, потом может генерироваться форма поискового запроса, только потом мы можем добраться до списка кандидатов. Достоинством данного подхода является его универсальность и относительная сочетаемость со стандартными реализационными схемами Web-интерфейсов и оконных приложений.

Третьим вариантом является использование технологии drag and drop для установления прямых ссылок между айтемами. Смысл такого установления заключается в том, что мы ассоциированный айтем можем найти или породить независимым интерфейсом, потом «взять» его и «положить» в требуемую ячейки или прямо на текущий айтем. Положительным моментом такого подхода является использование базовых представлений о модели структуризации: то, что есть сущности (айтемы) и есть отношения и процесс редактирования заключается в поиске или вводе сущностей, редактировании их атрибутов и установлению отношений между различными сущностями. Тем не менее, модель редактирования данных, пользователю непривычна и иногда от пользователя требуются знания о деталях используемой онтологии.

Возможно, наиболее естественной для пользователя, была бы модель текстового ввода сущности, на которую устанавливается ссылка. Например, в поле «место проживания», можно было бы позволить указать некоторый текст с минимальными элементами формального синтаксиса, скажем: г. Новосибирск. Система смогла бы разобрать такую запись с помощью словарей понятий и сокращений и найти соответствующую сущность в базе данных для использования в качестве ссылки, а в случае отсутствия – ввести новый элемент и установить на него ссылку. Понятно, что при отождествлении возмож-

на альтернативность, но об этом, можно предупредить и дать варианты для выбора. Такая схема требует развитой базы данных по основным категориям сущностей, указаний текущего контекста редактирования и системы соглашений о применяемых служебных словах и сокращениях или качественной системы анализа естественно-языкового текста, основанной на алгоритмах искусственного интеллекта. Кроме того, эта схема не является универсальной. Например, если речь идет о документах (фотодокументах и др.), то поисковое действе вряд ли будет базироваться на имени документа. Также сложно идентифицировать на словесном уровне мероприятия, а иногда и организации.

Вышеупомянутые особенности редактирования семантических сетей, позволяют сделать вывод об отсутствии привычных пользовательских моделей для выполнения поисковых и редактирующих действий. Возможно, нужно предусматривать наличие альтернативных способов для получения того же результата.

Довольно часто проблемой становится не отсутствие информации, а ее избыток. Например, относительно конкретной персоны в базу данных могут быть занесены тысячи фото и других документов, в которых он отражен, сотни документов, в которых он автор, десятки конференций, в работе которых он участвовал и т.д. Целостное изображение такого информационного портрета приводит к трудностям в нахождении пользователем нужной информации и связей между информационными единицами. Для приведенного примера, целью пользователя может являться нахождение номера телефона данной персоны, что может оказаться непростым занятием при неудачном представлении упоминавшегося ранее списка айтемов, ссылающихся на данный.

Таким образом, речь идет об ограничении видимости, задаваемой статически, при настройке конкретного интерфейса или динамически, при работе с конкретным объектом. Мы опробовали два подхода для решения данной проблемы. Первый заключается в определении множеств классов и свойств, которые в данный момент перестают рассматриваться. Эти множества задаются группами, так называемыми темами. Например, тема «документы» включает в себя несколько классов документов, а также ряд сложных отношений и их свойств, связанных с документами. Это – авторство, отражение, публикация. В интерфейсе работы с данными есть настроечный список тем, в котором можно отметить те темы, которые будут использоваться при просмотре или редактировании данных и те, которые использоваться не будут.

Другой способ связан с введением информационной значимости айтемов. Этот субъективно вводимый фактор, дает возможность списки айтемов сортировать по некоторому критерию важности и оставлять в формируемом информационном портрете только самую существенную информацию.

5 Исторические информационные системы

Был сформулирован и обоснован подход к созданию электронных архивов. Суть этого подхода заключается в том, что порождаются электронные образы единиц архивирования, создается или используется база данных традиционных сущностей (персоны, организации, географические точки, события), осуществляется «привязка» единиц архивирования к элементам базы данных, создается информационная система, позволяющая осуществлять навигацию и поиск по базе данных.

В Институте систем информатики СО РАН, в течение 10 лет ведутся исследования и выполняются разработки по созданию информационных систем исторической направленности. В частности, разработан и уже несколько лет эксплуатируется электронный архив документов А.П. Ершова. В результате выполненных работ, были сформулированы основные принципы создания фактографических систем исторической направленности: система должна отражать факты реального мира; для этого используются формальные системы, позволяющие через спецификации определять номенклатуру объектов отражения и основные свойства данных; необходимо создавать информационные системы распределенного типа; базы данных должны быть контекстно независимыми; должны активно использоваться международные стандарты; интерфейсами к системам должны быть для людей Web-клиенты (браузеры), а для программных агентов – Web-сервисы.

Наиболее крупным проектом последнего периода времени явился проект фотоархива Сибирского отделения http://soran1957.ru, запуск в эксплуатацию первой очереди которого был приурочен к 50-летию СО РАН. Архив в настоящее время содержит около 15 тыс. фотодокументов, база данных охватывает период с 1957 года и содержит информацию о 9 тыс. персон, 2 тыс. организаций и событий, также в базу данных помещена информация о членах Российской академии наук за весь период ее функционирования, взятая из базы данных РАН.

Архив представляет собой программно-аппаратный комплекс, реализующий сканирование и другую обработку фотоматериалов, обеспечивающий сохранность и защиту информации от несанкционированного доступа, предоставляющий средства редактирования базы данных, рабочие и публичные интерфейсы. Публичный интерфейс архива был запущен в опытную эксплуатацию в 2007 году, за первые три месяца эксплуатации, его посетило около 20 тысяч посетителей, пользователями просмотрено более 600 тысяч страниц.

Институт систем информатики СО РАН будет продолжать работу над фотоархивом как в части технологии, так и в части пополнения новыми данными, подробного описания фотодокументов, исправления неточностей. На этой же платформе предполагается создавать и другие корпоративные и

личные архивы содержащие базу фактов, а также документы, фото, видео, аудио документы, описание людей, организаций, событий и проектов.

Литература

[1] Berners-Lee Tim, Hendler James, Lassila Ora, The Semantic Web. In Scientific American, volume 284(5), pages 34–43, 2001.

[2] Марчук А.Г. Распределенные электронные архивы, библиотеки и базы данных. Препринт 122, Институт систем информатики им. А.П. Ершова СО РАН, Новосибирск, 25 с., 2004.

[3] Online community for the Universal Description, Discovery, and Integration OASIS Standard — http://uddi.xml.org

[4] The Dublin Core Metadata Initiative — http://dublincore.org/

[5] The CIMI Profile. Release 1.0H. A Z39.50 Profile for Cultural Heritage Information — http://www.cimi.org/public_docs/HarmonizedProfile/HarmonProfile1.htm

[6] Когаловский М.Р. Перспективные технологии информационных систем, М.: ИТ-Экономика, 2003. 288 с.

[7] Web Ontology Language (OWL). — http://www.w3.org/2004/OWL

[8] Resource Description Framework (RDF). — http://www.w3.org/RDF

[9] FOAF — http://www.foaf-project.org/

About Distributed Factographic Systems

Alexander Marchuk

In this research work special model is proposed for fixation and structuring of historical facts. The model is based on factographic principles of collection of formally specified data and information. The structuring model includes several classes of instances such as: persons, organizing systems, geographic systems, documents and some other. Instances are connected by simple or complex relations. Proposed approach is oriented to processes, upcoming in time and space, it allows to create databases, which does not become obsolete and does not depend upon time, place and position of perceiving person. Factographic approach can be used in a set of such applications as: building of digital archives, museums, historical encyclopedia and directories. Same ideas can be efficiently implemented in small systems for office work organizing. New possibilities in use of structured field of facts for analytics and knowledge extraction are under investigation. Specific features of the proposed approach are: main concept is from Semantic Web, new ontology of non-specific entities is proposed, strong orientation on distributed databases, use of the idea of information space of documents. This approach was partially implemented in applied projects of A.P. Ershov Institute of Informatics Systems.