

Конечность вычислений обосновывается следующими соображениями. Правила «склейки» соответствуют отождествлению вершин графа запроса, а их число конечно. Новые фрагменты (которых не было в исходной ситуации), пригодные для применения правил «склейки», могут получиться только в результате применения правил доопределения. Если применяются только правила доопределения, то рано или поздно между всеми или только некоторыми вершинами графа запроса будут установлены все допустимые связи, оговоренные в онтологии. Если происходит «переключение» между этими двумя видами правил, то при каждом таком переключении количество вершин уменьшается и (или) количество ребер увеличивается. Этот процесс не может длиться бесконечно в силу конечности числа вершин и количества допустимых связей между парой вершин в онтологии.

Интеграция данных на основе онтологий для обеспечения информационной поддержки управленческих решений

П. А. Ломов, М. Г. Шишаев

Институт информатики КНЦ РАН, Апатиты

1. Введение

В наше время принятие своевременных и обоснованных решений в процессе управления какой-либо макросистемой, будь то социально-экономическая, банковская, государственная система становится возможным только на основе обработки больших объемов различной информации с использованием информационных технологий и моделирования возможных последствий реализации принимаемых решений. Выполнение данной задачи предполагает создание единого унифицированного интерфейса для некоторой совокупности независимых источников данных, которыми могут выступать традиционные системы баз данных, Web-сайты, файлы структурированных данных. Одной из важных проблем при этом является наличие того факта, что, как правило, обрабатываемые источники данных являются гетерогенными. Их гетерогенность проявляется на различных уровнях: системном, синтаксическом, структурном, семантическом, что является следствием использования различных программных и аппаратных архитектур, на которых реализованы информационные системы, а также применении различных форматов и моделей представления данных. Необходимо также отметить, что множество источников может также расширяться со временем. Все это делает проблему интеграции данных довольно сложной и многоуровневой, для решения которой следует в обязательном порядке принимать во внимание, как структурные и синтаксические различия информационных ресурсов, так и семантику их отдельных информационных элементов, для обеспечения смысловой интероперабельности данных и разрешения семантических конфликтов.

2. Современные подходы к интеграции

Проблема интеграции данных характеризуется большим разнообразием постановок задач, подходов и методов, используемых для их решения. Благодаря использованию кроссплатформенных технологий, таких как CORBA и DCOM удастся решить проблему системной гетерогенности, возникающей вследствие различных программно аппаратных платформ, на которых реализованы информационные системы. Появление и распространение языка XML позволяет задать общий синтаксис для представления информации, тем самым избавиться от синтаксических различий. Однако, до сих пор не до конца решенными остаются проблемы преодоления логической и, еще в большей степени, семантической неоднородности источников данных.

В общем случае можно выделить два способа интеграции — материализованная и виртуальная. В случае материализованной интеграции данных создается новый источник интегрированных данных, который используется автономно от породивших его источников. При необходимости его состояние приходится синхронизироваться с их актуальным состоянием.

В случае виртуальной интеграции, система интеграции использует виртуальный источник, который в любой момент времени «содержит» актуальные данные интегрируемых источников, и синхронизации его состояния не требуется. Данные для ответа на запросы пользователя, в этом случае берутся из информационных источников непосредственно в процессе выполнения запроса. Права владельцев исходных интегрируемых источников сохраняются. Они продолжают автономно поддерживать их в своих интересах, предоставляя вместе с тем права доступа к их ресурсам пользователям системы интеграции данных в соответствии с установленным регламентом. Авторизованные пользователи системы интеграции получают непосредственный доступ только к виртуальному источнику, воплощаемому данной системой.

Среди архитектур систем с виртуальной интеграцией, наиболее популярной является архитектура медиаторов и адаптеров. Медиатор, в данном случае, это функциональный компонент системы, задачей которого является поддержка некой глобальной схемы и организации обработки пользовательских запросов, выраженных в ее терминах, декомпозируя их на подзапросы, адресуемые соответствующим источникам. Адаптеры служат для представления информационных ресурсов источников в терминах глобальной модели данных, принимают на обработку подзапросы от медиатора, выполняют обработку источника и возвращают результаты медиатору.

В контексте сопоставления глобальной схемы и схем источников применяют следующие подходы: *Global as View* и *Local as View*. Первая из них (*Global as View*) предусматривает определение глобальной схемы в терминах схем локальных источников. Такой подход более эффективен в случае, когда множество всех используемых источников предопределено. При использовании второй разновидности рассматриваемой архитектуры (*Local as View*) предполагается, что схема для каждого из локальных источников данных определяется в терминах глобальной схемы. Хотя в этом случае усложняется отображение пользовательских запросов в среду локальных источников данных, такой подход имеет важное достоинство — он допускает динамичность состава множества интегрируемых источников данных.

Помимо общих подходов к интеграции выделяют также более специфичные подходы с использованием онтологических описаний, направленных в первую очередь на решение проблемы семантической гетерогенности. Выделяют следующие подходы: подход с использованием единой онтологии, мульти-онтологический и гибридный подходы. Наиболее универсальным является последний из перечисленных, так он позволяет довольно просто добавлять новые информационные ресурсы, также в нем отсутствует необходимость установления связей между терминами разных онтологий, как в случае применения мульти-онтологического подхода. Таким образом, его можно использовать для интеграции информационных ресурсов родственных предметных областей с возможностью последующего добавления информационных ресурсов в пространство интеграции. Главным недостатком этого подхода является невозможность использования уже существующих онтологий, термины которых не описаны в разделяемом словаре, а это приведет к неполному или усредненному отражению семантики информационных элементов.

3. Подход с использованием общего расширяемого тезауруса

Для устранения основного недостатка гибридного подхода предлагается использовать в качестве общего не статический словарь, а расширяемый тезаурус. Основной целью данного метода является предоставить возможность создавать независимую онтологию для каждого информационного источника, позволяющую точно отразить его семантику, при этом термины данной онтологии должны добавляться в расширяемый тезаурус и связываться с уже хранящимися там терминами. Каждый термин будет связан с той онтологией, в которой он присутствует, что позволит,

используя связи, производить трансляцию запроса в термины конкретной онтологии. Данный подход, в сущности, предполагает производство интеграции онтологий на уровне установления частичного соответствия онтологий, а не совместимости или унификации, что существенно облегчает данную задачу, но в тоже время позволит осуществлять поиск информации достаточно эффективно.

4. Архитектура и порядок работы системы

Система, представленная далее, ориентирована на преодоление логической и, главным образом, семантической гетерогенности динамически — на этапе исполнения. Ее можно отнести к типам системам с «виртуальной» интеграцией, основанной на использовании медиаторов и реализующей разновидность гибридного подхода, заключающейся в использовании расширяемого тезауруса, вместо статичного общего словаря. Ее основным отличием от классической модели с использованием медиаторов является использование двухуровневого адаптера информационного источника, включающего агента, выполняющего перевод запроса из интегрированной схемы в термины локального семантического представления, и контент адаптера, осуществляющего отображение объектов семантического представления на реальные объекты информационного ресурса. Таким образом, можно выделить два уровня интеграции: семантический, реализуемый агентом, и структурный, реализуемый контент адаптером. Структура системы включает следующие функциональные компоненты (рис. 1):

- пользовательский интерфейс — представляет формы пользовательского запроса, взаимодействует с семантическим хранилищем, для предложения объектов поиска пользователю, осуществляет передачу запроса системе выполнения пользовательских запросов;
- система выполнения пользовательских запросов — производит отбор агентов в семантическом хранилище, генерирует задания для агентов и выполняет их активизацию, также осуществляет извлечение результатов задания и передача их пользовательскому интерфейсу;
- семантическое хранилище — служит для взаимообмена информацией между агентами, осуществляет кэширование популярных запросов, хранит информацию об агентах и их задачах и результатах работы агентов;
- агент — выполняет поиск в семантическом представлении ресурса, передает результаты выполнения запроса в семантическое хранилище;

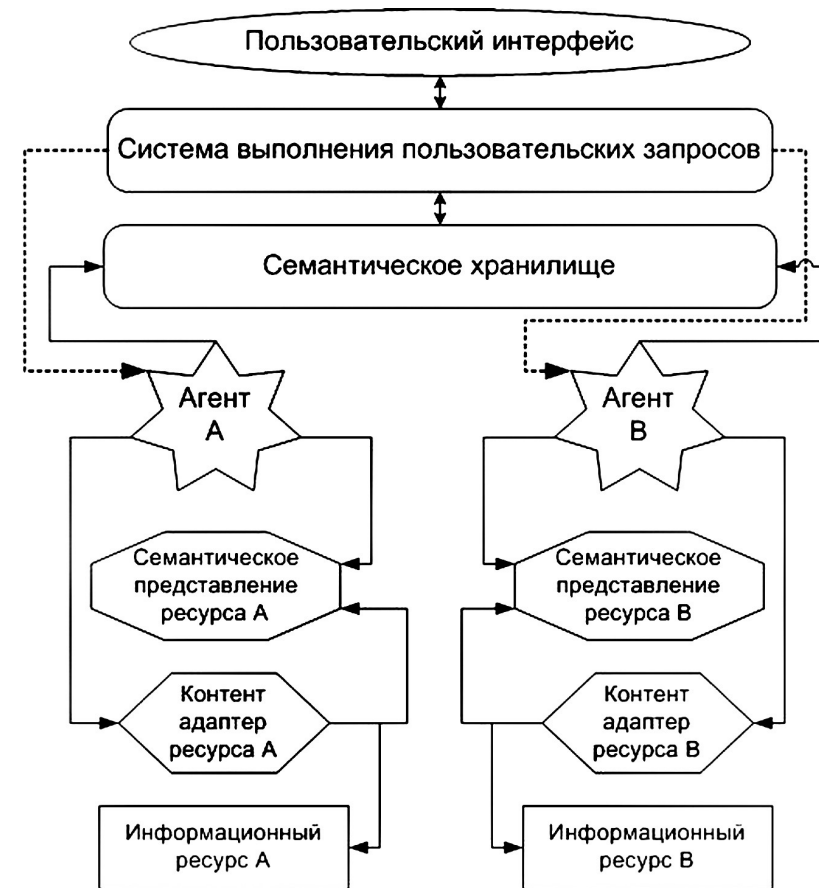


Рис. 1. Структура системы семантической интеграции

- семантическое представление ресурса — хранит машинопонимаемую метаинформацию, отражающую семантику реального информационного ресурса;
- контент адаптер ресурса — выполняет трансляцию объектов семантического представления в объекты реального информационного ресурса, также может выполнять копирование объектов информационного ресурса в его семантическое представление;
- информационный ресурс — текстовый документ, база данных, электронная таблица и т. д.

Использование отдельных агентов для одного или нескольких информационных ресурсов позволяет, локализовать в рамках области поиска одного агента, и, соответственно, упростить задачу совместного использования нескольких семантически связанных онтологий, применяемых в качестве семантических представлений ресурсов. Это делает возможным использование любого из существующих подходов отображения терминов одной онтологии в другую или, при возникновении непреодолимых трудностей, создание отдельных поисковых агентов для определенных ресурсов.

Общий порядок работы системы можно разделить на три основных этапа:

1. Формулировка запроса пользователя в терминах семантического хранилища.
2. Выбор агентов для выполнения пользовательского запроса.
3. Поиск агентами информации в семантических представлениях ресурсов.

Для начала функционирования системы необходимо, чтобы в семантическом хранилище хотя бы один агент прошел процедуру регистрации, которая описана далее. В процессе работы предполагается регистрация новых агентов, для расширения области поиска.

Целью первого этапа является интерактивная формулировка пользовательского запроса, для точной ее трансляции в термины словаря семантического хранилища, так как именно это, впоследствии, будет играть важную роль в получении результатов, в достаточной степени релевантных той информации, что хотел найти пользователь.

Простым и эффективным способом получения точного смысла запроса является его создание в интерактивном режиме с помощью конструктора запросов, что позволяет сразу обозначить объект поиска в терминах, присутствующих в семантическом хранилище. Для приемлемого функционирования конструктора необходимо выполнение следующих условий:

Онтология семантического хранилища должна обладать достаточно полным набором экземпляров класса «объект» и класса «свойство», которые включает термины, свойства и их синонимы;

В случае неоднозначной трактовки пользовательского запроса пользователю представлялось несколько интерпретаций, среди которых он имел бы возможность выбрать более подходящую с его точки зрения. Например, если пользователь ищет автомобиль по цене менее 50 000 руб. и указывает при этом термин «машина» в качестве объекта поиска, то система должна вывести несколько вариантов запроса, где указаны более конкретные понятия, такие как «стиральная машина», «посудомоечная машина», «автомобиль».

Рис. 2. Форма ввода пользовательского запроса

Минусом этого решения может являться накладывание определенных ограничений на структуру запроса.

Запрос пользователя будет состоять из элементов — конструкторов. Конструктор может быть как наименованием термина предметной области, объекта реального мира, какого-либо свойства, так и идентификатором экземпляра соответствующего класса в онтологии семантического хранилища. Также он может содержать значение свойства без явного определения типа.

Сама процедура построения запроса состоит из задания конструкторов, каждому из которых соответствует одно поле на форме запроса (рис. 2).

Рассмотрим работу системы на первом этапе по шагам:

1. Пользователь начинает вводить первый конструктор запроса. По мере заполнения поля конструктор, в свою очередь, будет искать в семантическом хранилище экземпляры класса объектов, со сходным значением свойства «наименование» для внесения его в список, который будет представлен пользователю.
2. Пользователь должен выбрать объект из списка, представленного конструктором. В случае если желаемый объект в списке отсутствует, пользователь должен вводить его синонимы, до тех пор, пока он не будет представлен в списке. Например, при поиске автомобиля по определенной цене, если в ответ на ввод слова «машина» конструктор представил пустой список, необходимо осуществить ввод синонимов: автомобиль, автомашина. Если даже после ввода синонимов конструктор не находит подходящих экземпляров в классе

объектов, то это говорит либо об отсутствии агентов, осуществляющих поиск данных сущностей, либо о недостаточно полном представлении списка синонимов.

3. Ввод второго конструкта осуществляется с целью конкретизации поиска и происходит аналогично вводу первого с той лишь разницей, что в список подстановки для второго конструкта попадают лишь наименования свойств, связанных с экземпляром класса объектов, введенного в качестве первого конструкта, а также экземпляров, имеющих с первым связь типа синоним.
4. Вводом третьего конструкта, пользователь определяет значение или ограничение свойства, заданного на предыдущем шаге.
5. Пользователь может либо завершить формирование запроса, либо ввести еще несколько конструктов, являющихся наименованием свойств и значениями этих свойств.
6. Пользователь отправляет запрос.

Таким образом, конструктор является частью системы выполнения пользовательских запросов, осуществляющую работу с конечным пользователем и семантическим хранилищем для формирования пользовательского запроса. Следует также заметить, что в запросе при каждом вводе наименований экземпляров класса объектов или свойств, в конструкты попадают идентификаторы экземпляров, а не их наименования, что позволяет избежать в дальнейшем дополнительных действий по их поиску в онтологии семантического хранилища. Содержимое третьего же конструкта, определяющего значение или ограничение свойства, не изменяется.

После получения пользовательского запроса на втором этапе система выполнения пользовательских запросов производит отбор агентов для осуществления поиска:

1. Среди экземпляров класса связей, имеющих тип — синоним, осуществляется поиск тех, у которых одно из значений свойств, указывающих на экземпляры, входящие в связь, равно первому идентификатору экземпляра в запросе. Из найденных извлекаются идентификаторы экземпляров-синонимов, и затем по значению свойств этих экземпляров, указывающих на экземпляры агентов, предоставивших данные синонимы, система выполнения пользовательских запросов создает экземпляр класса заданий, связывая его с экземпляром класса агентов, которому адресована данная задача, и экземпляром класса объектов, вызвавшим ее создание.
2. Далее рассматривается второй и третий конструкт запроса, определяющий соответственно экземпляр класса свойств и бестиповое

значение. Аналогично предыдущему шагу, ищутся экземпляры класса свойств, являющихся синонимами и связанными с экземплярами класса объектов-синонимов, найденных на предыдущем шаге. Для каждого найденного свойства, предоставленного определенным агентом, создается экземпляр класса значений, в который заносится содержание третьего конструкта, и который связывается через свои свойства с соответствующими экземплярами класса заданий и класса свойств.

3. Система выполнения пользовательских запросов заносит в некоторый список идентификаторы созданных экземпляров класса задач и соответствующие им идентификаторы агентов. Далее происходит рассылка запросов агентам.

Таким образом, на данном этапе происходит трансляция, на основании семантических связей, заданных в семантическом хранилище, объектов пользовательского запроса в объекты локальных семантических представлений. Следует отметить, что этот процесс происходит с сохранением семантики исходного пользовательского запроса.

Последний этап работы системы заключается в поиске агентом информации в семантическом представлении, обслуживаемым им информационным ресурсом:

1. Агент получает запрос и обращается к семантическому хранилищу для определения, через идентификатор экземпляра адресованной ему задачи, наименования экземпляра объекта для поиска, а также наименований экземпляров свойств и значений этих свойств.
2. Далее, используя полученные данные задачи, агент выполняет запрос к семантическому представлению и получает идентификатор экземпляра искомого объекта в семантическом представлении, а через него — значение свойства, определяющего универсальный идентификатор ресурса. Данный идентификатор и является ответом на запрос пользователя, который агент, сохраняет в семантическом хранилище, в качестве значения свойства-результата в экземпляре задачи, и передает уведомление системе о завершении работы.
3. После получения уведомлений система выполнения пользовательских запросов предоставляет пользователю универсальные идентификаторы результатов, используя которые пользователь может с ними ознакомиться.

Следует сразу отметить, что реализация универсального идентификатора может быть абсолютно разной и зависит от адаптера ресурса. Например, если адаптер представляет собой набор CGI-скриптов, то универсальным идентификатором в данном случае будет некая веб-ссылка,

иницирующая выборку определенного контента из реального информационного ресурса. Аналогично логика взаимодействия адаптера и агента может быть иной, это зависит от структуры и формата информационного ресурса. Например, агент, найдя экземпляр в онтологии, являющейся семантическим представлением ресурса, хранящего информацию о научных статьях и их атрибутах, может сначала дать команду, являющуюся значением определенного свойства данного экземпляра, адаптеру. Тот, в свою очередь выполнит определенную обработку, скажем, особым образом размеченных текстовых документов, сформирует в результате временный документ формата html и передаст веб-ссылку на него агенту, в качестве универсального идентификатора.

5. Процедура регистрации агента в семантическом хранилище

Процедура регистрации производится с целью сбора информации об агентах и тех терминах и свойствах, по которым они могут производить поиск в обслуживаемых ими информационных ресурсах. Эта информация впоследствии используется конструктором для формулировки пользовательского запроса, помимо этого система выполнения пользовательских запросов применяет ее для активации того или иного агента для произведения поиска. Хранение регистрационной информации осуществляется в онтологии, где существуют следующие классы: агент, термин, свойство, связь, задача.

В ходе регистрации агент предоставляет свои идентификационные данные, а также список наименований классов, свойств онтологий тех ресурсов, с которыми он работает. Помимо этого агент может внести список связей между предоставленными терминами, которые должны быть явно включены в семантическое хранилище. Типы связей задаются в семантическом хранилище, и пока предполагается использовать типы: синоним, антоним, ассоциация, в случае необходимости список типов связей может быть расширен. Эти данные добавляются в онтологию семантического хранилища с указанием ссылки на представивший их агент.

6. Обеспечение контроля доступа к данным

Важным аспектом системы интеграции данных является обеспечение контроля доступа к данным, которая состоит в определении прав доступа пользователей и агентов к определенным информационным элементам информационных ресурсов. Для решения этой проблемы необходимо

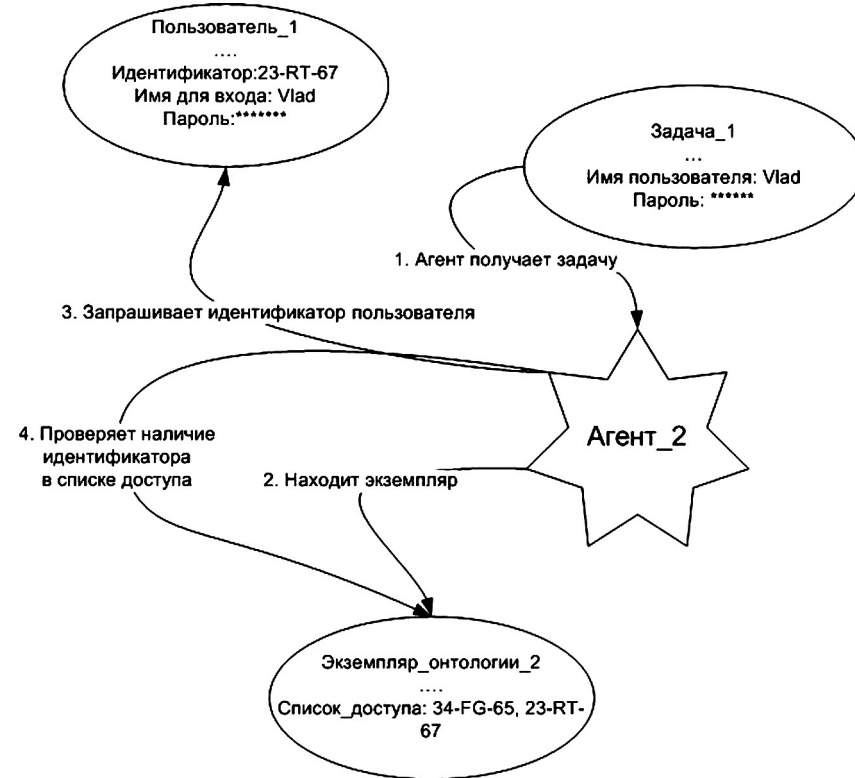


Рис. 3. Проверка прав доступа пользователя

обозначить процедуры идентификации и аутентификации субъектов, взаимодействующих с системой, определить реестр для хранения идентификационной информации и процедуру проверки прав доступа (рис. 3).

Идентификатор пользователя будет проверяться всем множеством агентов, поэтому его необходимо хранить в семантическом хранилище в виде значения свойства экземпляра класса «пользователь». Таким образом, можно определить процедуру идентификации, как создание соответствующего экземпляра с заданием значений обязательных свойств: пароля, имени для входа и идентификатора. Процедура аутентификации, соответственно, будет состоять в проверке соответствия введенного пароля имени для входа. В случае успешного прохождения аутентификации все экземпляры класса «задача», создание которых он инициирует, будут содержать имя для входа и пароль в качестве значений своих свойств.

Агенты, в свою очередь, могут использовать идентификаторы пользователей из семантического хранилища для определения уровня доступа к экземплярам своих онтологий, путем определения значений особых свойств, например, список записи и список чтения. Соответственно, агент, перед выполнением задачи, получает идентификатор пользователя из семантического хранилища, используя имя для входа и пароль, указанные в ней.

Хранение прав доступа внутри информационного элемента позволяет также контролировать обращение агентов друг к другу, в этом случае агент рассматривается в качестве обычного пользователя.

7. Онтология семантического хранилища

В этом разделе описывается онтология, включающая в себя информацию об основных объектах, используемых в ходе работы поисковой системы и связях, между ними. Структура онтологии представляется следующим образом (рис. 4).

Назначение и свойства классов онтологии:

- агент — определяет агента, прошедшего регистрацию в семантическом хранилище, хранит идентификатор, ссылку для обращения, указывает на термины и свойства, предоставленные агентом и на задачи, созданные для него;

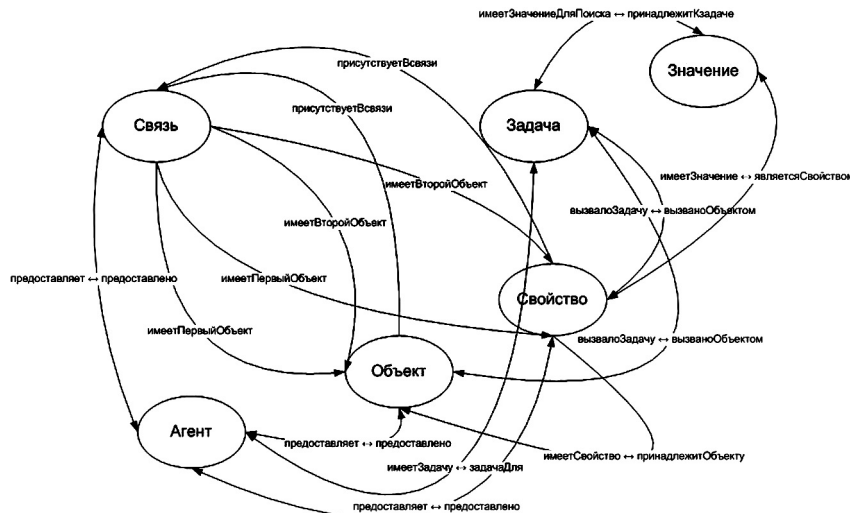


Рис. 4. Онтология семантического хранилища

- объект — сущность, предоставленная определенным агентом, которую он может найти в семантических представлениях, обслуживаемых им. Также она указывает на связанные с ним свойства, на определенную связь с другим термином, на агента, предоставившего его, на задачу, которую инициировало присутствие данного термина в запросе;
- свойство — некий признак, характеризующий термин, имеет свойства, аналогичные классу терминов. Кроме этого он указывает на экземпляры класса значений, которые определяют его в задачах;
- значение — определяет значение, введенное пользователем, которое будет использоваться агентом при поиске в семантическом представлении, указывает на задачу и на свойство, значением которого оно является;
- задача — каждый экземпляр данного класса определяет задачу для конкретного агента, имеет идентификатор, указывает на агента и инициирующие эту задачу термин или свойство и его значение свойства;
- связь — определяет связь между двумя терминами, указывает на термины или свойства и агента, предоставившего его.

Классы: агент, связь и задача также имеют свойство полезность, отражающее соответственно оценку эффективности выполненных поисков данным агентом, частоту использования определенной связи и релеванность результатов конкретной задачи. На основании этих оценок система корректирует алгоритм своей работы с целью использования более результативных агентов и сокращения объемов поиска в семантическом хранилище.

8. Заключение

Произведение семантической интеграции, так или иначе, предполагает произведение интеграции онтологий, которые в свою очередь, описывая одну и ту же или перекрывающиеся области интереса, могут существенно отличаться степенью детализации, а также подходом к построению концептуализации предметной области. Исходя из этого, для решения проблемы семантической гетерогенности широкого круга информационных источников, необходимо обеспечить как возможности полного отражения их семантики посредством задания произвольных онтологий, так и установления отношений между терминами различных онтологий. Подход, представленных в данной работе, предоставляет решение этих задач. Однако, впоследствии, вероятно возникнет проблема, связанная с возникновением семантических конфликтов, заключающихся

в появлении сходных по наименованию, но разных по смыслу терминов, или противоречащих друг другу связях между терминами. Данные трудности предполагается разрешить с помощью варьирования показателя полезности, того или иного термина или связи. Предполагается, что оценка будет производиться во время процедуры, включающей анализ полученных результатов задач и их пользовательских оценок, отсюда будет определяться степень полезности конкретного агента, вес определенной связи между терминами. Валидность связи также будет зависеть от количества ее повторений в списках связей, предоставляемых агентами на этапе регистрации, которые можно рассматривать как подтверждения ее правильности.

Необходимо заметить, что данная система может производить поиск информации об объекте и его характеристиках, содержащихся в различных информационных ресурсах. Примером такого запроса может служить поиск информации о конкретном автомобиле, имеющим определенную цену, причем модель должна быть максимально безопасной. Необходимая информация может находиться в двух разных по своей направленности семантических представлениях — одно содержит каталог автомобилей на продажу, а другая специализируется на хранении информации о моделях, результатов тестов их характеристик. Система в данном случае формирует две задачи, первая из которых направлена на поиск моделей с требуемым уровнем безопасности, а затем найденные результаты используются в качестве условий второй, адресованной уже другому агенту. Эта возможность является довольно ценной особенно в контексте использования системы для одной предметной области, где информационные ресурсы описывают одинаковые объекты, но с различных позиций и степени детализации.

Семантическое хранилище может также быть использовано для обмена данными между агентами с целью пополнения своих семантических представлений, генерации новых информационных ресурсов, содержащих совокупную информацию. Для достижения этой цели планируется определить особый тип, так называемых, кросс агентных задач. Также необходимо разработать общие принципы взаимодействия агентов, для того чтобы можно было не только осуществлять поиск по запросу пользователя, но и совместную обработку информации.

В дальнейшем будет также необходимо формализовать представления основных объектов системы интеграции и основные процедуры, такие как поиск и регистрация агента, добавления термина в тезаурус. Это во много облегчит дальнейшую разработку программного прототипа системы интеграции.

Литература

1. *Ломов П. А., Шишаев М. Г.* Интеграция семантически связанных информационных ресурсов на основе онтологий для эффективного информационного обеспечения рационального природопользования // *Ломов П. А., Шишаев М. Г.* Глубокая переработка минеральных ресурсов: Сборник материалов IV школы молодых ученых и специалистов «Сбалансированное природопользование» (6–8 ноября 2007 г.). Апатиты: Изд-во КНЦ РАН, 2008. С. 243–247.
2. *Bianchini D., De V. Antonellis* «Ontology-based Integration for Sharing Knowledge over the Web». [Электронный ресурс] http://www.doc.ic.ac.uk/~pjm/diweb2004/DIWeb2004_Part8.pdf
3. *Petrenko M., Jamil H.* Query Ontologies for Autonomous Online Resource Integration Systems. [Электронный ресурс] <http://www.cs.wayne.edu/~max/docs/QueryOntologies.pdf>
4. *Бездушный А. Н., Меденников А. М., Серебряков В. А.* Подход к интеграции информационных коллекций в ИСИР РАН. [Электронный ресурс] <http://dbserv.ihep.su/~pubs/aconf00/dconf00/ps/056.pdf>
5. *Клещев А. С., Артемьева И. Л.* Математические модели онтологий предметных областей. Часть 3. Сравнение разных классов моделей онтологий // Научно-техническая информация. Сер. 2 Информационные процессы и системы. № 4. 2001. С. 10–15.
6. *Калиниченко Л. А.* Методология организации решения задач над множественными распределенными неоднородными источниками информации. [Электронный ресурс] <http://synthesis.ipi.ac.ru/synthesis/publications/itedu05/itedu05.pdf>
7. *Скворцов Н. А.* Применение уточнения понятий в решении задач манипулирования онтологиями. [Электронный ресурс] http://rcdl2007.pereslavl.ru/papers/paper_70_v1.pdf
8. *Коголовский М. Р.* Тенденции развития технологий управления информационными ресурсами в электронных библиотеках. [Электронный ресурс] http://www.rcdl2006.uniyar.ac.ru/papers/paper_98_v1.pdf