УДК 004.4'423

Амонс А. А., к.т.н.,
Резник Д. И.,
Скарга С. А.,
Островский С. М.

УНИФИКАЦИЯ ИСТОЧНИКОВ ДАННЫХ В РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ ПОСРЕДСТВОМ КОНТЕКСТНО-СВОБОДНЫХ ГРАММАТИК

Национальный технический университет Украины
«Киевский политехнический институт»

Предложен подход к решению задачи унификации множества разнородных источников данных при помощи синтаксического преобразования запросов на основе модифицированных контекстно-свободных грамматик. Подход имеет применение в распределенных информационных системах с децентрализованным и централизованным управлениями. Разработана структурная схема унифицированного источника данных и алгоритмы синтаксической унификации запросов.

Введение

Большинство сложных программных комплексов в современном мире взаимодействуют лишь с одним выбранным источником данных. Подобная практика стала стандартом де факто в современной индустрии программного обеспечения. Использование и специализация приложений под конкретный вид источников данных объясняется следующим:

- Использование разнородных источников влечет за собой накладные расходы на разработку: изучение спектра предполагаемых источников данных, написание модулей интеграции, взаимосвязь источников данных;
- Меньшая универсализация системы позволяет ее сделать более простой и дешевой в обслуживании;
- Упрощение и уменьшение количества интерфейсов взаимодействия системы с окружающим миром упрощает дальнейшую разработку и развитие системы.

Наиболее распространенными и широкоиспользуемыми на данный момент являются реляционные источники данных. Однако, специализируясь на отдельных типах источников данных, информационные системы ограничиваются специфическим набором приложений подобных источников и их функциональными возможностями, таким образом ограничивая собственные характеристики и возможности [1].

Существуют отдельные классы задач, требующие использования унифицированных источников данных. К примеру, поисковые машины и приложения семантических сетей основывают свою работу на использовании распределенных и разнотипных источников данных. Подобный подход позволяет выбирать наиболее релевантные, важные и современные данные [2].

Под унифицированными источниками данных будем понимать объединения множеств источников данных в логически единый источник данных, способный выполнять необходимые операции извлечения и поиска информации с помощью определенного механизма доступа среди объединенных источников [2].

Нередко также встает проблема создания современных информационных систем с обязательным требованием интеграции с предшествующими системами, которые использовали устаревшие механизмы хранения данных. Обеспечение подобной интеграции обыкновенно является задачей отдельного программного модуля [3].

Однако использование унифицированных источников данных обладает рядом несомненных преимуществ.

Прежде всего, с увеличением количества используемых источников данных увеличивается суммарный объема данных, используемых информационной системой. Таким образом информационная система приобретает преимущество перед конкурирующими системами за счет качественно другого уровня выдаваемых данных.

Дополнительно повышается масштабируемость информационной системы за счет добавления дополнительных источников данных в программный комплекс. Подобное действие

может выполняться динамически в процессе работы информационной системы, таким образом уменьшая накладные расходы на обслуживание.

Интеграция разнородных источников данных позволяет увеличить функциональные возможности информационной системы с помощью использования встроенных алгоритмов в источниках данных. Реляционные источники данных могут предоставлять возможности создания динамических представлений, агрегированных выборок; в то время как объектно-ориентированные источники данных позволяют формировать собственные типы и классы данных.

Применение унифицированных источников данных позволяет повысить качество выдаваемой информации благодаря тому, что информация уточняется и дополняется в различных источниках. Качественный анализ однородных данных из различных источников позволит получить наиболее полное представление об объекте анализа.

Еще одним преимуществом является повышение производительности системы за счет использования распределенных запросов к источникам данных. Традиционно, большинство реляционных источников данных поддерживает параллельный режим работы, в том числе и для больших массивов данных, однако он имеет свои ограничения на аппаратном уровне (пропускная способность жесткого диска, объем оперативной памяти, вычислительная мощность ЦПУ). Использование распределенных источников данных позволяет одновременно запрашивать различные данные без значительных потерь в скорости обработки запросов [3].

Проблемам упрощения и универсализации доступа к гетерогенным источникам данных посвящена данная статья.

Рассмотрим основные подходы к построению программных комплексов с использованием унифицированных источников данных.

Анализ существующих решений

В работах [4-6] рассматривается подход к созданию подобных информационных систем с использованием т.н. «оберток» или «посредников». Для каждого отдельного источника данных дополнительно в систему вводится модуль интеграции (в литературе такие модули часто называют «обертка», «посредник», «интегратор», «гомогенизатор», «драйвер»). Перед такими модулями ставят следующие задачи:

– Преобразование запросов к источнику данных в вид, понятный источнику. Т.е. происходит синтаксическое согласование между информационной системой и источником данных;

– Преобразование параметрических величин в вид, воспринимаемый источником данных. На данном этапе происходит лексическое согласование информационной системы и источника данных;

– Преобразование набора результатов в вид, ожидаемый информационной системой в качестве ответа – семантическое преобразование.

Соответственно, для каждого источника данных создается определенный модуль, решающий указанные задачи. Написание и интеграция в системе подобных модулей не всегда является тривиальной задачей и требует значительных ресурсозатрат. Положение осложняется тем, что интеграционный модуль не всегда есть физическая возможность реализовать.

Еще одной проблемой, связанной с интеграционными модулями, является необходимость дополнительной сборки и развертывания информационной системы с целью подключения модулей, т. е. теряется прозрачность и интерактивность системы для конечного пользователя.

В работе [7] показано усовершенствование данного метода, основанное на идее разбиения модуля интеграции на две составляющие: гомогенизатор и интегратор. В задачи первого входит преобразование входных и выходных значений в определенную форму, заранее оговоренную и заданную требованиями к программному комплексу. Второй модуль является шаблонным, и отвечает за подключение гомогенизаторов к информационной системе.

В данном случае облегчается программистская модель, повышается качество исходных кодов информационной системы. Однако, в связи с гомогенизацией источников данных теряется специфический функционал источников данных, который как правило позволяет повысить скорость работы системы.

Общая модель подобной системы представлена на рис. 1

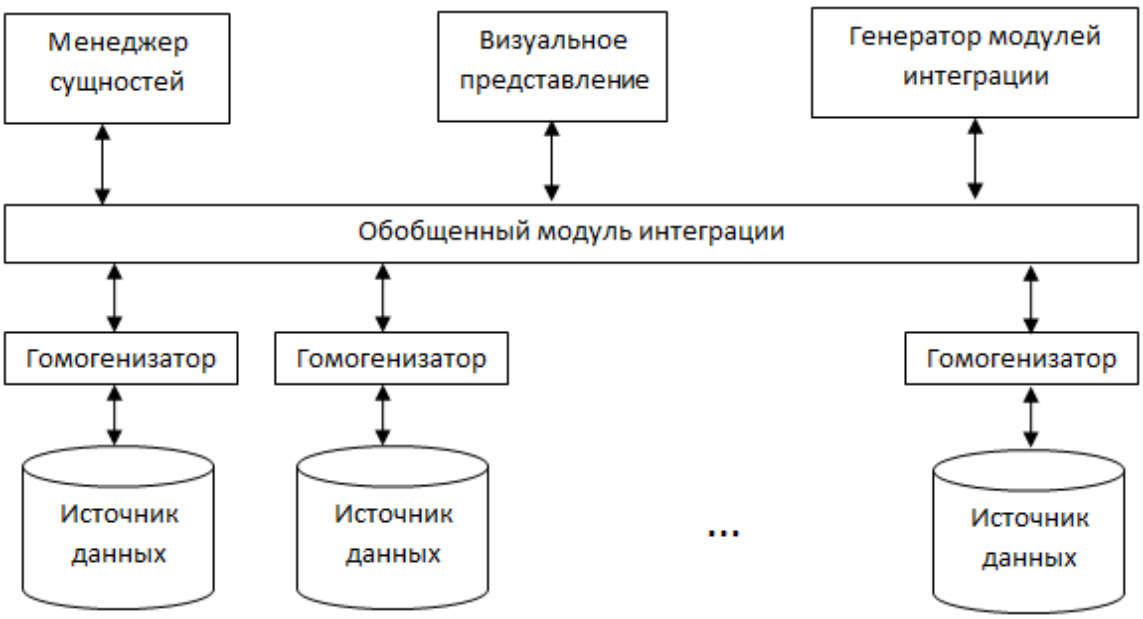

Рис. 1. Общая модель системы с использованием интеграторов и гомогенизаторов

В разработанной информационной системе [8] для устранения указанных проблем предлагается использование единой согласованной синтаксической формы данных, представленной в формате *XML*. Определяются правила преобразования, согласно которым наборы данных и запросы преобразовываются в согласованную форму. Различают два вида правил: преобразования от источнику к приемнику и от приемника к источнику. Правила описываются в формате *XML* и используются для трансляции данных с целью интеграции источников данных.

Подобная система имеет несколько неоспоримых преимуществ перед предыдущим подходом: модули интеграции имеют шаблонный характер и могут быть созданы динамически без вмешательства разработчиков; система правил имеет расширяемую и простую природу; внедрение новых источников данных требует значительно меньше усилий.

Однако в данной информационной системе используется функционально ограниченный промежуточный согласовывающий формат описания данных и непрозрачный механизм преобразования.

В работе [9] описан подход с использованием централизованного хранилища представлений и возможностей источников данных, которые являются упрощенной разновидностью оберток. Преимуществами такого подхода является простота реализации и использования.

В работах [10-11] использована обобщенная структура представления данных и запросов (в качестве такой структуры выбран *XML*). Использование однородной, обобщенной и распространенной структуры позволяет встраивать оптимизационные модули в начале процесса обработки запроса, что уменьшает время отклика системы и увеличивает быстродействие, а также позволяет унифицировать представление данных во всей системе.

Обобщая можно сделать вывод, что все современные методы построения информационных систем с использованием гетерогенных источников данных имеют комплексную модульную структуру, сложность которой возрастает нелинейно с увеличением количества разнородных источников данных.

На данный момент выработаны общие механизмы взаимодействия с гетерогенными источниками данных, которые трудоемки в реализации [6, 9-10].

Результаты анализа представлены в таблице 1. Под знаком «+» в приведенной таблице следует понимать наличие определенной возможности, и наоборот для знака «-» – отсутствие соответствующей возможности. Согласно представленным в таблице данным видно, что ни одна система не обладает достаточным спектром возможностей для применения в распределенных информационных системах и скорее представляют набор компромиссов, на которые приходится идти для внедрения унифицированных источников

данных. В частности, ни один из подходов не обладает возможностями распределенного выполнения и обработки запросов, что создаст большую нагрузку на каналы связи в информационной системе. Большую сложность также представляет возможность дальнейшего расширения информационной системы, что увеличивает расходы на обслуживание.

Таблица 1. Сравнительный анализ средств унификации источников данных.

Качество \ Система	Оберточные представления	Интеграторы и гомогенизаторы	Правилоосновванная систе-	Хранилища метаданных	*XQuery, XML*
Расширяемость	трудоемко	трудоемко	+	трудоемко	трудоемко
Централизованное использование	+	+	+	+	+
Децентрализованное использование	+	+	-	-	-
Возможность подключения модулей предварительной оптимизации	-	+	-	-	+
Возможность использования различных унифицированных моделей представления данных	+	+	+	-	-
Возможность распределенного выполнения/обработки	-	-	-	-	-
Возможность подключения новых источников данных «на лету»	-	-	+	-	+
Возможность полноценного использования всего доступного функционала источника данных	+	-	-	+	+
Возможность синтаксического анализа запроса на источнике данных	-	-	-	-	-

Использование контекстно-свободных грамматик в задаче унификации источников данных

Задачу взаимодействия с гетерогенными источниками данных можно рассматривать как две отдельные задачи: задача преобразования результирующих массивов данных в воспринимаемую форму и задачу преобразования входных запросов в понятный источнику данных вид. Поскольку задача преобразования результирующих массивов данных носит более технический характер и не является ключевой для проблемы унификации источников данных, в статье она рассматриваться не будет. Проанализируем задачу преобразования запросов подробнее.

Преобразование входных запросов можно расценивать как проблему преобразования текста запроса из заданного языка запросов в язык запросов источника данных. Подобная проблема имеет большое сходство с проблемой перевода естественных языков, однако с некоторыми ограничениями: имеется строгое определение синтаксиса — под строгим понимается существование четко определенных последовательностей ключевых слов, нарушение последовательности ведет к ошибке отработки запроса; ограниченный словарь синтаксических конструкций, измеримый не более чем в сотнях — для естественных языков он может состоять из более миллионов записей; каждой синтаксической единице противопоставляется единственное семантическое значение. Данная задача также очень сходна с задачей компиляции исходных кодов программных систем в исполняемые файлы – задача преобразования текста программы в машинный код, который воспринимается исполняющей средой.

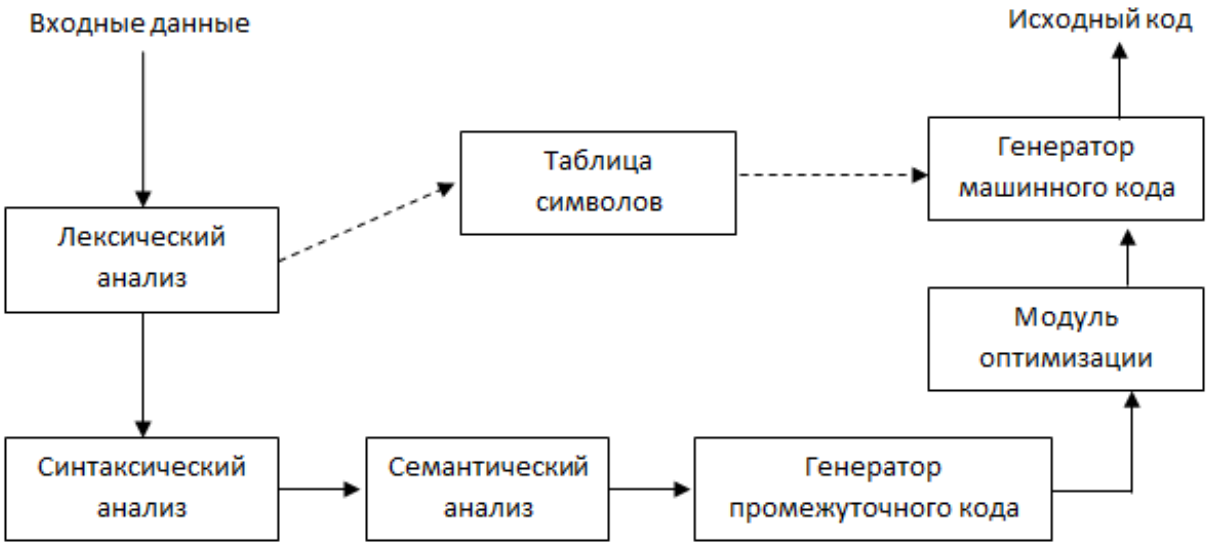

Рис. 2. Фазы компиляции

Ввиду указанных выше обстоятельств имеет смысл применить подходы из областей вычислительной лингвистики и построения компиляторов для формирования синтаксических преобразователей.

Представленный на рис. 2 процесс компиляции полностью соответствует поставленной задаче с некоторыми допущениями: входными данными является запрос в терминах исходного языка запросов; промежуточным представлением является унифицированный язык запросов, сконструированный для однозначного и однообразного описания функциональных возможностей поддерживаемых языков запросов; результирующим кодом является преобразованный запрос в терминах языка источника данных. В области машинного перевода данный подход называется переводом с использованием промежуточного языка (interlingua), когда грамматические конструкции переводятся из исходного естественного языка в язык-посредник, искусственный либо с некоторой точки зрения удобный для проведения дальнейшего перевода, и дальнейший перевод в целевой язык [12].

Синтаксис языка принято описывать в терминах *контекстно-свободных грамматик*.

Определение 1. Грамматика - это четверка G=(N, T, P, S), где N – алфавит нетерминальных символов; T – алфавит терминальных символов, N∩T=Ø; P – конечное множество правил вида α→β, где α∈(N∪T)*N(N∪T)*, β∈(N∪T)* (предикаты); S∈N – начальный символ (или аксиома) грамматики[12].

Определение 2. Пусть дана грамматика G=(N, T, P, S). Если каждое правило грамматики имеет вид α→β, где α∈N, β∈(N∪T)*, то ее называют грамматикой типа 2, или контекстно-свободной (КС-грамматикой) [12].

Необходимо также некоторым образом описать соотношение между терминами языка-источника и унифицированного языка запросов для дальнейшего преобразования в обобщенный вид.

Данную корреляцию целесообразно описывать внутри контекстно-свободной грамматики для упрощения структуры синтаксических преобразователей и уменьшения количества структур данных.

Для достижения этой цели внесем в предикаты дополнительный элемент – вторичное правило в терминах унифицированного языка запросов, определяющее одинаковую с исходным правилом семантику. Модифицированную таким образом грамматику будем называть *схемой синтаксического преобразования*. Получим следующее определение:

Определение 3. Схема синтаксического преобразования – это шестерка G=(N1, T1, N2, T2, P, S), где N1 – алфавит нетерминальных символов; T1 – алфавит терминальных символов, N∩T=Ø; N2 – унифицированный алфавит нетерминальных символов; T2 – унифицированный алфавит терминальных символов; P – конечное множество правил вида α→β|γ→δ, где α∈(N∪T)*N(N∪T)*, β∈(N∪T)* (предикаты); S∈N – начальный символ (или аксиома) грамматики.

С использованием указанных выше определений, сформулируем общий алгоритм преобразования синтаксиса запроса.

Алгоритм 1. Общий алгоритм работы синтаксического преобразователя.

Лексический анализ входного запроса, в процессе которого переменные запроса заменяются специальными маркерами, а сами переменные заносятся в таблицу вида «маркер – переменная» – таким образом достигается лексическая независимость запроса.

Синтаксический анализ входного запроса – согласно схеме синтаксического преобразования (далее – ССП) строится абстрактное синтаксическое дерево запроса, представленное в виде неориентированного графа.

Преобразование в промежуточный синтаксис выполняется согласно вторичным предикатам ССП: терминалы в вершинах графа заменяются эквивалентными терминалами контекстно-свободной грамматики унифицированного языка запросов. При необходимости происходит изменение структуры графа – удаляются и добавляются ребра между соответствующими вершинами графа.

Преобразование в синтаксис языка источника данных – аналогично п. 3 за исключением того, что преобразование происходит из унифицированного языка запросов в язык запросов источника данных.

При формировании конечного запроса происходит нисходящее прохождение по абстрактному синтаксическому дереву, в результате которого терминалы и переменные в вершинах графа устанавливаются на синтаксически правильное положение в запросе.

Рассмотрим детальнее полный процесс преобразования синтаксиса между языками запросов.

Алгоритм 2. Преобразование синтаксиса.

Шаг 1. На вход преобразователя поступает запрос определенного синтаксиса в виде строки.

Шаг 2. По входной строке определяется соответствующая ССП поочередной проверки строки на соответствие предикатам каждой из доступных схем. В случае, если найдено больше одной ССП, выбирается первая из доступных. В случае, если не найдено ни одной подходящей схемы, алгоритм завершается.

Шаг 3. Построение синтаксического дерева. Входной строке ставится в соответствие множество предикатов в порядке их использования. Левая часть предиката становится вершиной верхнего уровня в дереве, правая часть предиката становится вершиной нижнего уровня в дереве, и между данными вершинами устанавливаются ребра.

Шаг 4. Согласно множеству предикатов, сформированному на шаге 3 данного алгоритма, формируется множество предикатов унифицированного языка запросов, полученные из ССП.

Шаг 5. Происходит нисходящее прохождение по дереву с заменой соответствующих терминалов в вершинах графа. В случае, если терминал находится в недопустимом месте, он изымается из дерева и помещается во временное хранилище элементов графа с указанием терминала, к которому данный элемент принадлежит. В случае, если по прохождению данный элемент не найден, он формируется в соответствующем месте графа, и к нему примыкает элемент из промежуточного хранилища. Из промежуточного хранилища он при этом удаляется.

Шаг 6. Шаги 4-5 повторяются с другой ССП.

Шаг 7. Происходит нисходящее прохождение по дереву слева направо. Все значения в пройденных вершинах добавляются в выходную строку.

Шаг 8. Происходит синтаксический анализ выходной строки на соответствие ССП источника данных.

На рис. 3 приведена структура интерфейса доступа к унифицированному источнику данных с использованием контекстно-свободных грамматик. На вход интерфейса передается запрос к определенному источнику данных в синтаксисе языка из списка поддерживаемых данной системой. Поддерживаемые языки источников данных определяются набором ССП, хранящихся в базе метаданных системы. При поступлении запроса он анализируется в модуле синтаксического анализа с целью нахождения соответствующих ССП языка запроса и языка источника данных. Запрос вместе с ССП передаются на модуль преобразования запроса, где он преобразовывается согласно алгоритму 2.

Преобразованный запрос может быть дополнительно оптимизирован и проверен на совместимость с выбранным источником данных модулями оптимизации запросов и синтаксической проверки соответственно. В

дальнейшем общий для всех источников данных модуль выполнения запросов выполняет обращение к источнику данных с преобразованным запросом и передает результирующие данные на модуль выходного преобразования. В задачи последнего входит преобразования результирующих массивов данных в унифицированный для всех систем вид. Таким образом происходит цикл обработки и выполнения запроса на унифицированном источнике данных (УИД).

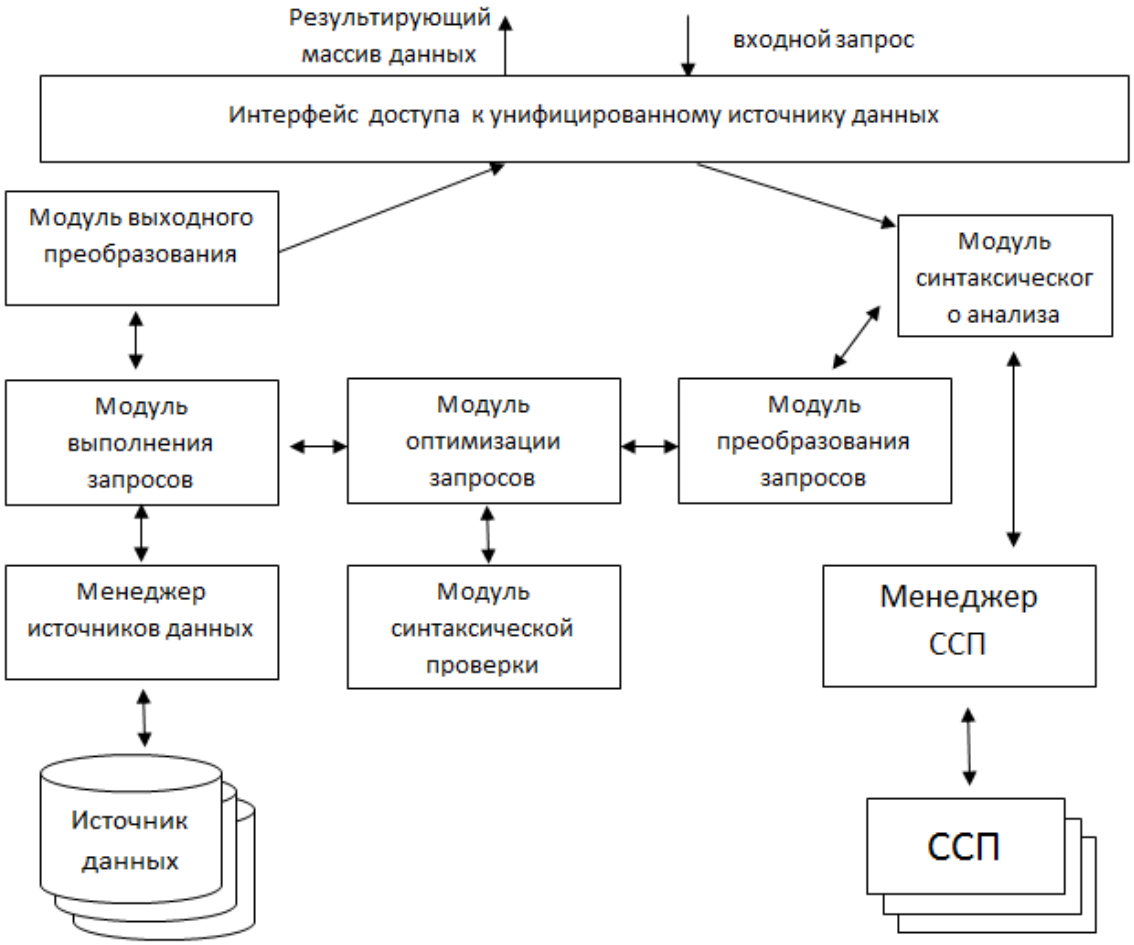

Рис. 3. Структурная схема интерфейса доступа к данным отдельного узла в распределенной информационной системе с использованием контекстно-свободных грамматик

С точки зрения распределенных информационных систем, как показано на рис. 4, можно выделить два сценария использования приведенных выше УИД: децентрализованный и централизованный.

В случае децентрализованного использования УИД каждый умел имеет собственный УИД, включающий определенные ССП и источники данных. Соответственно, каждый узел имеет возможность обращаться к источникам данных независимо от остальных узлов; в то же время каждый узел имеет возможность взаимодействовать с источниками данных других узлов посредством унифицированных запросов. Дополнительно в такую систему вводится узел синхронизации ССП на случай, если входящий с другого узла запрос имеет неизвестный синтаксис.

В случае централизованного использования УИД рассматриваемая система имеет вид «ведущий» – «ведомый». Определяется узел, имеющий доступ к источникам данных, через который передаются запросы остальных узлов. Запросы могут поступать в различном синтаксисе и преобразовываются на ведущем узле в необходимый. Ведомые узлы не имеют самостоятельной возможности получать доступ к данным.

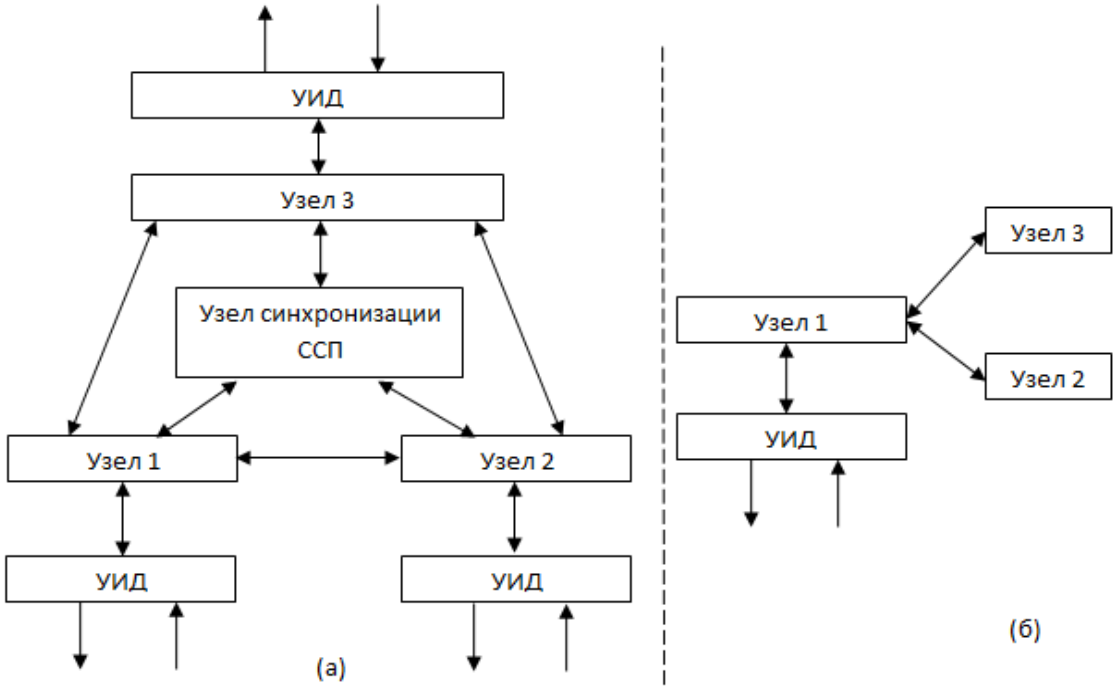

Рис. 4. Сценарии использования унифицированного источника данных:
а) децентрализованный сценарий; б) централизованный сценарий.

Выводы

В статье представлен подход к преобразованию синтаксиса языков запросов источников данных при помощи схем синтаксического преобразования, которые основываются на контекстно-свободных грамматиках. Представленный подход использует промежуточный язык запросов, что позволяет значительно уменьшить количество необходимых контекстно-свободных грамматик для преобразования синтаксиса.

Использование схем синтаксического преобразования обладает рядом преимуществ:

- Интуитивно простое и понятное представление синтаксиса языка запросов источника данных;
- Простая расширяемость и дополняемость – для расширения/уточнения синтаксиса достаточно добавить в грамматику соответствующие предикаты;
- Производительное и несложное преобразование формы;
- Возможность децентрализованного и централизованного использования;
- Повышенная масштабируемость количества источников данных;
- Использован развитый математический аппарат и наработанные практики из области построения компиляторов и машинного перевода;
- Использование промежуточного языка уменьшает количество необходимых преобразователей – от $n \times m$ до $n+m$.

Остались не рассмотренными проблемы согласования схем данных между источниками данных, формирующих унифицированный источник данных, а также проблема преобразования результирующих массивов данных в унифицированный вид. Данные проблемы являются темами дальнейших исследований.

Список литературы

1. I.F. Cruz. Ontology driven data integration in heterogeneous networks. / I.F. Cruz, H. Xiao // Studies in Computational Intelligence. Complex Systems in Knowledge-based Environments: Theory, Models and Applications. – 2009. – № 168. – P. 75–97.

2. L. Cao. Agent mining: The synergy of agents and data mining. / L. Cao, V. Gorodetsky, P. A. Mitkas. // IEEE intelligent systems. – 2009. – vol.24, № 3. – P. 64–72.

3. G. Di Lorenzo. Data integration in mashups / G. Di Lorenzo, H. Hacid, H. Paik // SIGMOD Record. – 2009. - vol 38, № 1. – P. 59–66.

4. Кашников А.В. Интеграция гетерогенных источников данных на основе рекурсивной декомпозиции. / А.В. Кашников, Л.Н. Лядова // International Journal

«Information Technologies Knowledge». – 2011. – Т. 5, № 3. – С. 274-284.

5. S. Chawathe. The TSIMMIS project: Integration of heterogeneous information sources. / S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, J. Widom // Proceedings of the 16th Meeting of the Information Processing Society of Japan. – 1994. – P. 7–18.

6. A. Langegger. A semantic web middleware for virtual data integration on the web. / A. Langegger, W. Wöß, M. Blöchl // 5th European Semantic Web Conference. – 2008. – P. 493-507.

7. L.L. Yan. Accessing heterogeneous data through homogenization and integration mediators. / L.L. Yan, M. Tamer, O.L. Liu // Proceedings of the Second IFCIS International Conference on Cooperative Information Systems. – 1997. – P. 130–139.

8. R. Shaker. A rule driven bi-directional translation system for remapping queries and result sets between a mediated schema and heterogeneous data sources. / R. Shakerl, P. Mork, M.S.2, M. Barclayl, P. Tarczy-Homoch, M.D. // Proceedings of the AMIA Symposium. – 2002. – P. 692–696.

9. Ганопольский Р.М. Представление знаний в гетерогенных распределенных базах данных на примере ИНГРИС ТюмГУ. / Р.М. Ганополоьский, Д.Б. Кепещук // Вестник кибернетики. – 2006. – № 5. – С. 70–76.

10. M.I. Ali. DeXIN – an extensible framework for distributed XQuery over heterogeneous data sources. / M.I. Ali, R. Pichler, H.L. Truong, S. Dustdar // Proceedings of International Conference on Enterprise Information Systems. – 2009. – P. 1–12.

11. I. Manolescu. Answering XML queries over heterogeneous data sources. / I. Manolescu, D. Florescu, D. Kossmann // Proceedings of 27th International Conference on Very Large Data Bases. – 2001. – P. 241–250.

12. Д. Хопкрофт. Введение в теорию автоматов, языков и вычислений / Д. Хопкрофт, Р. Мотвани, Д. Ульман; пер. с англ. А. Ставровского. – М.: «Вильямс», 2008. – 528 с.