

Рассредоточение реализации приложений в распределенной среде предметных посредников *

© А.Е. Вовченко

ИПИ РАН
itsnein@gmail.com

Аннотация

В статье обсуждается проблема рассредоточения реализации алгоритма решения задач в среде предметных посредников. Задача возникает в связи с тем, что спецификация реализации алгоритма решения задачи может быть задана на различных компонентах среды предметных посредников - в виде программ на языке программирования, на языке правил предметных посредников, и в виде взглядов. В работе предложена модель рассредоточения для многоязычных программ, а также предложен подход построения квази-оптимального рассредоточения.

1. Введение

В различных областях науки наблюдается экспоненциальный рост объема получаемых экспериментальных (наблюдательных) данных. Например, в астрономии текущий и ожидаемый темп роста данных от наземных и космических телескопов удваивается в течение периода от шести месяцев до одного года. Сложность использования таких данных увеличивается еще и вследствие их естественной разнородности. Число организаций, получающих данные наблюдений в отдельных областях науки в мире, велико. Разнообразие (информационная несогласованность) получаемой информации вызывается, в частности, не только большим числом организаций, производящих наблюдения, и их независимостью, но и разнообразием объектов наблюдения, непрерывным и быстрым совершенствованием техники наблюдений, вызывающим адекватные изменения структуры и содержания накапливаемой информации. Это приводит к необходимости использования неоднородной, распределенной информации, накопленной в течение значительного периода наблюдений технологически различными инструментами.

Основной идеей в инфраструктуре доступа к

множественным неоднородным информационным ресурсам является введение промежуточного слоя между ресурсами, и потребителями информации. Основными компонентами промежуточного слоя являются предметные посредники [1], существующие независимо от информационных ресурсов. Использование предметных посредников для неоднородных информационных ресурсов предоставляет интеллектуальный подход к интеграции информации.

Архитектура промежуточного слоя предметных посредников рассматривается как основная для решения научных задач над множеством интегрируемых неоднородных распределенных информационных ресурсов. Реализован подход [1], при котором для класса приложений формируется спецификация предметной области посредников независимо от существующих информационных ресурсов. Также важной частью среды для решения задач является сопряжение средств поддержки посредников с языками программирования (ЯП), что позволяет существенно расширить возможности пользователей при решении задач.

Вместе с тем возникает проблема реализации спецификации решаемой задачи. Проблема возникает в связи с тем, что каждый компонент среды предметных посредников обладает широкими возможностями, которые зачастую пересекаются, что приводит к неоднозначности выбора конкретной реализации. Например, какая-то часть спецификации реализации алгоритма решения задачи может быть реализована в виде программы на ЯП, либо в виде программы к посреднику, и.т.д. Существует ряд компонентов среды предметных посредников, между которыми может быть рассредоточена реализация приложения:

- Системы программирования
- Программа к посреднику на языке правил
- GLAV [6] взгляды, определяемые при регистрации ресурсов
- Программируемые адаптеры ресурсов данных и сервисов
- Информационные ресурсы

Построение эффективного рассредоточения реализации встает особенно остро, когда требуется многократный прогон решаемой задачи. Например, в астрономии, некоторая задача была сформулирована для площадки размером 1

Труды 13^й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» - RCDL'2011, Воронеж, Россия, 2011.

квадратный градус. Тогда чтобы прогнать решаемую задачу по всему небу для полосы

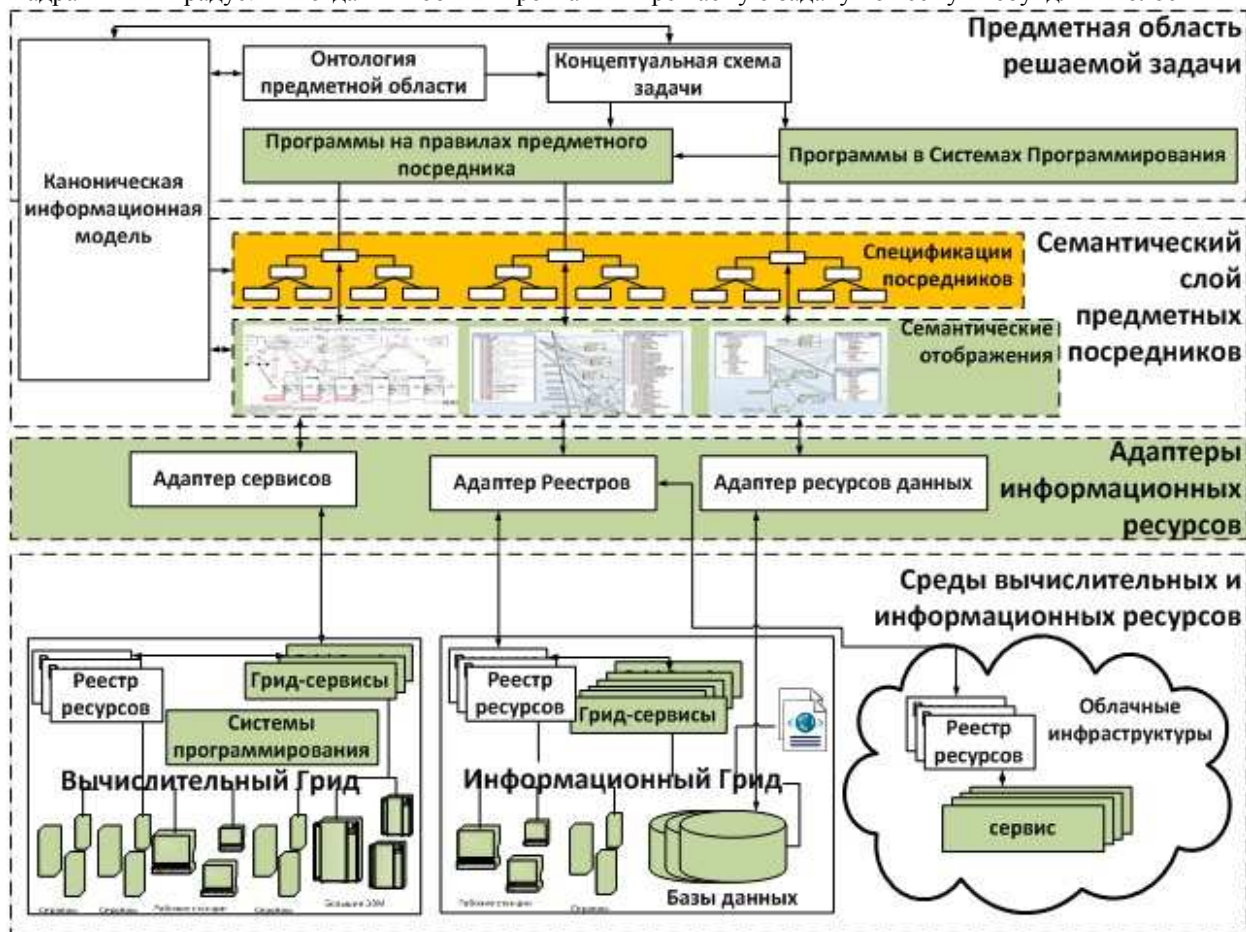


Рисунок 1. Обобщенная архитектура среды предметных посредников для решения задач

шириной 1 градус, нам потребуется 360 прогонов. Для того чтобы прогнать решаемую задачу для всего неба потребуется $360 \cdot 180 = 64800$ запусков. При таких условиях даже выигрыш в скорости выполнения одной прогонки на 1 минуту, экономит полтора месяца вычислений.

В работе исследуются проблемы построения эффективного рассредоточения реализации приложений в рамках промежуточного слоя предметных посредников для решения научных задач.

В следующем разделе представлена обобщенная архитектура среды предметных посредников. Далее приведен обзор существующих подходов. Далее рассматривается определение рассредоточения в среде предметных посредников, а также функций оценки рассредоточения. Затем представлен подход к построению квази-оптимального рассредоточения в среде посредников.

2. Обобщенная архитектура среды предметных посредников

Исходя из принципов построения сред предметных посредников, разработана обобщенная

архитектура среды предметных посредников для решения задач, представленная на рисунке 1.

Архитектура предметных посредников включает:

- уровень информационных ресурсов, включающий базы данных, сервисы, средства программирования, и другие средства, возможно организованные в рамках грид, облачных или других инфраструктур;
- слой адаптеров, обеспечивающий интероперабельность ресурсов благодаря технической унификации их интерфейсов и введению дистанционных механизмов обращения к ресурсам (адаптеры, осуществляющие преобразование запросов, выраженных в канонической информационной модели посредников, в их представление в информационной модели ресурса);
- уровень предметных посредников, каждый из которых создает спецификацию предметной области для решения некоторого класса задач, используя каноническую информационную модель («эсперанто») для представления семантики предметной области и унифицированного отображения разнообразных видов информационных моделей ресурсов (моделей

данных, сервисных моделей, онтологических моделей, процессных моделей);

- уровень задач (приложений), формулируемых в терминах одного или нескольких посредников.

Для решения задач используется метод, движимый приложениями. Отправляясь от предметной области задачи, определяется онтология предметной области (понятия и связи между ними), строится концептуальная схема предметной области, содержащая информационные структуры и методы, необходимые для решения задачи. Таким образом, образуется семантическая спецификация решения задачи, полностью независимая от конкретных ресурсов. В терминах концептуальной схемы предметной области формулируются программы для решения задачи на языке правил посредника. Кроме того формулируются программы и на языках программирования. После этого определяются инфраструктуры содержащие ресурсы, необходимые для решения задачи. Далее, на уровне ресурсов идентифицируются ресурсы, релевантные задаче, используя реестры доступных инфраструктур. Релевантные задаче ресурсы регистрируются в предметных посредниках, образуя тем самым семантические отображения классов ресурсов в классы посредника (взгляды). Для всех ресурсов релевантных задаче автоматически или с участием эксперта генерируются адаптеры, а также задаются способности адаптеров и ресурсов.

3. Обзор существующих подходов

В том виде, в котором задача поставлена в настоящей работе, она ранее не рассматривалась. Тем не менее, можно выделить ряд областей, которые в той или иной степени близки к поставленной задаче. В первую очередь задача близка к подходам построения эффективных планов выполнения запросов в среде предметных посредников. Во вторых, задача близка к работам, посвященным анализу многоязычных программ (программ заданных на нескольких языках программирования). Наконец, задача близка к вопросам оптимизации выполнения потоков работ. Рассмотрим отдельно эти различные аспекты.

В работе [7] рассматривается подход к оптимизации запросов, который можно обобщить и на процесс построения минимального рассредоточения. Так, для построения рассредоточения необходимо: определить пространство поиска (возможные варианты рассредоточений), и далее использовать либо оценочные модели, либо статистику.

В работе [4] обсуждаются вопросы планирования запросов в среде предметных посредников. В качестве подхода к планированию используется общая парадигма планирования, называемая планированием посредством переписывания (Planning by Rewriting – PbR) [5]. В работе [4] оцениваются параметры выполнения запроса для построения эффективного плана.

Главная идея подхода заключается в том, что строится начальный план, а затем данный план улучшается посредством применения набора описанных декларативных правил. Аналогичный подход используется в настоящей работе при поиске рассредоточения на основании экспертных правил и рассматривается в следующем разделе.

В работе [12] исследуется проблема построения плана выполнения запроса, в частности, особая роль уделяется проблеме выполнения операции join в среде предметных посредников. В [12] используется простая оценочная модель, подсчитывающая общее число подзапросов, что дает весьма общую оценку. В подходе поиска минимального рассредоточения в качестве оценочной модели используется более объективная – временная характеристика. В работе [12] предлагается два алгоритма построения плана. Первый – «жадный» алгоритм, строит быстро достаточно эффективный, но возможно не самый оптимальный план. Второй алгоритм существенно сложнее, строит оптимальный план, но за существенно большее время. Аналогичный подход применен при построении минимального рассредоточения. Используются два алгоритма для построения рассредоточения. Первый – на основании экспертных правил – строит не всегда самое эффективно рассредоточение. Второй алгоритм полного перебора строит всегда минимальное рассредоточение.

В работе [3] исследуется проблема оптимизации запросов. В ней предполагается, что применение подхода на основе оценочной модели не всегда просто, т.к. нет доступа к статистике ресурсов и сложно оценить производительность ресурсов, поэтому предлагается кэшировать статистическую информацию реальных запросов к информационным ресурсам. Подобный подход применен при построении минимального рассредоточения. На основании статистической информации были выявлены экспертные правила для построения рассредоточений. Также в работе [3] кэшируются результаты запросов, что неприменимо в среде предметных посредников, т.к. ресурсы могут все время изменяться и важно найти минимальное рассредоточение независимо от данных. В случае же кэширования результата, можно быстро получить результат из кэша при не самом оптимальном рассредоточении.

В работе [10] обсуждается проблема анализа программ в многоязычных системах. Проблема возникает в связи с тем, что программа может быть специфицирована одновременно на разных языках, при этом возможности языков могут зачастую пересекаться. Наиболее распространенный подход для анализа программ на разных языках – это отображение программы во множество общих структур, в рамках которых уже осуществляется анализ. В различных проектах в качестве общих структур используются ERM-модели, либо много чаще используются концептуальные графы [9, 11].

Недостатком подхода является его ориентация на анализ конкретных программ. Иными словами – общие структуры языков в виде концептуального графа строятся для конкретных программ, а не для языков в целом. При поиске минимального рассредоточения для работы с многоязычной спецификацией алгоритма решения задачи используется граф (модель рассредоточения). При этом важным моментом является то, что функциональные возможности разных языков определены изначально и не зависят от конкретной модели рассредоточения и конкретной спецификации реализации алгоритма решения задачи, то же самое справедливо и для экспертных правил.

Наконец задача построения рассредоточений близка к работам по оптимизации потоков работ [8], представленных графами, т.к. минимальное рассредоточение строится путем перестановок операций в графе (модели рассредоточения). В работе выявлен важный принцип подобных перестановок. При перестановке операций в графе важно доказать эквивалентность преобразований. Сам подход, рассматриваемый в работе [8], представляет собой последовательное применение правил оптимизации графа, что близко к применению экспертных правил при построении рассредоточения.

4. Постановка задачи рассредоточения в среде предметных посредников

Обобщенная архитектура среды предметных посредников для решения научных задач представлена выше на рисунке 1. Не нарушая общности рассуждений относительно рассредоточения реализации, архитектуру среды предметных посредников можно свести к тем компонентам среды, на которых возможно осуществить рассредоточение реализации алгоритма решения научной задачи. Реализация задается в виде программ и спецификаций следующих компонентов среды:

- программ на языках программирования;
- предметных посредников, специфицированных на языке правил;
- средств поддержки отображений классов ресурсов в классы посредников и языка определения отображений (взглядов);
- адаптеров информационных ресурсов и средств их программирования и определения их способностей;
- конкретных информационных ресурсов.

Определение 1. Реализацией алгоритма решения научной задачи или реализацией задачи в среде предметных посредников будем называть интероперабельную композицию следующих компонентов, порождаемых спецификациями: предметных посредников, программ, порожденных системами программирования, взглядов, адаптеров, ресурсов.

Реализация алгоритма решения задачи может быть задана не единственным образом. Возможности каждого из компонентов ограничены их функциональными возможностями. При этом различаются компоненты спецификации (КС) реализации задач и исполнительные компоненты (ИК). К компонентам спецификации относятся - ЯП, язык правил, язык определения взглядов, язык спецификации способностей адаптеров. К исполнительным компонентам – интерпретатор систем программирования (ИСП), исполнительный слой предметных посредников, адаптеры информационных ресурсов, информационные ресурсы.

Определение 2. Функциональной операцией будем называть конструкцию программы (спецификации), которой можно манипулировать в процессе построения рассредоточения как атомарной единицей.

Например, процедуры или функции в ЯП могут рассматриваться как атомарные операции с набором входных и выходных параметров.

Определение 3. Если функциональная операция $op1$ среди входных параметров имеет выходные параметры функциональной операции $op2$, то говорят, что операция $op1$ зависит от операции $op2$.

Стоит отметить, что у каждой операции может быть любое число зависимых от нее операций, и она может зависеть от любого числа операций.

Определение 4. Граф зависимостей функциональных операций или граф зависимостей – это ориентированный, вообще говоря, несвязный граф, без циклов, в вершинах которого расположены функциональные операции. Вершины в графе именуется так же, как и операции. Дуги в графе выражают зависимости операций, так что если операция $op1$ зависит от операции $op2$, то в графе зависимостей существует дуга, направленная от вершины $op1$ к вершине $op2$.

Реализация научной задачи, заданная как совокупность спецификаций на ЯП, на правилах и в виде взглядов взаимно однозначно отображается во внутреннее представление, представляющее собой граф зависимостей функциональных операций.

Определение 5. Назначением функциональной операции называется компонент, на котором операция специфицирована.

Например, для операций, заданных во взглядах, назначением являются средства поддержки отображений классов ресурсов в классы посредников, для операций, заданных в программе на ЯП, назначением является ЯП, для операций, заданных в программе на правилах, назначением является язык правил предметных посредников. Если две различные операции специфицированы на одном компоненте, то говорят, что у этих операций совпадает назначение.

Определение 6. Моделью рассредоточения будем называть граф зависимостей функциональных операций, для каждой операции которого

определены назначения. Также в модели рассредоточения определены возможные назначения для каждой из операций.

Состояние модели определяется назначением каждой из операций модели рассредоточения. При этом состояние непротиворечиво, если у зависимых операций совпадает назначение.

Для назначений заданы ограничения, определяющие корректность возможных назначений операций тем или иным компонентам (взгляды, правила, ЯП). Эти ограничения учитываются при генерации вариантов рассредоточения - переходе от одного варианта рассредоточения к другому (перестановке для краткости). Действия по перестановке, выполняемые при этом, должны удовлетворять ограничениям корректности назначений.

Определение 7. Рассредоточенной реализацией или рассредоточением будем называть некоторое непротиворечивое состояние модели рассредоточения.

При этом рассредоточение, получаемое непосредственно из текстовых спецификаций, называется начальным рассредоточением. Все операции, описанные во взглядах, в программе на правилах, в программе на ЯП получают в качестве назначения - язык спецификации взглядов, язык правил, ЯП соответственно.

Определение 8. Перестановкой операции op в модели рассредоточения будем называть такое изменение состояния модели, в котором у операции op изменяется назначение по сравнению с текущим, при этом назначение изменяется также и у всех операций, зависимых от операции op .

Переставляя операции, можно перебирать различные варианты рассредоточений. Несложно оценить и общее число вариантов рассредоточения (максимальное, в случае если нет зависимости операций). Пусть дано n - количество функциональных операций. Рассредоточение возможно между тремя компонентами - взглядами, правилами, ЯП. Тогда общее число вариантов рассредоточения для функциональных операций - 3^n .

В общем случае в исполнительном слое предметных посредников могут присутствовать компоненты, отвечающие за преобразование рассредоточения в план выполнения. В некоторых системах план выполнения может и не строиться, тогда множество функциональных операций совпадает с множеством исполнимых операций, а рассредоточение - с планом выполнения.

Определение 9. Планом выполнения называется дерево операций, необходимых для получения результата алгоритма решения научной задачи. Выполнение плана происходит от листьев к корню. Операции в плане выполнения называются исполняемыми.

Для наглядности рассмотрим два примера. Первый - это системы программирования. В общем случае для программы, написанной на ЯП, план

выполнения не строится, а происходит компиляция в машинный код, который уже исполняется. В данном случае программа на ЯП представляет собой как граф зависимостей функциональных операций, так и план выполнения. Второй пример - это СУБД. Для запроса на языке SQL строится план выполнения.

Стоит отметить, что для каждой операции в плане выполнения определено назначение.

Определение 10. Назначением исполняемых операции называется компонент, на котором операция исполняется.

План выполнения строится исходя из рассредоточения. В рассредоточении выбираются те операции, для которых назначения - взгляды, по множеству этих операций восстанавливаются текстовое представление взглядов. Аналогично восстанавливается текстовое представление программы на правилах. Программа переписывания запросов переписывает программу на правилах, используя взгляды. Планировщик планирует выполнение переписанной программы и строит план выполнения. Очевидно, что план может быть построен не единственным способом. Максимальное число вариантов плана выполнения, аналогично рассредоточению, - y^m , где y - количество компонентов, на которых происходит выполнение (учитываются все адаптеры, посредник, и интерпретаторы систем программирования), m - количество исполняемых операций.

Эффективность рассредоточения оценивается временем, затрачиваемым на выполнение задачи. Это время включает в себя:

- время построения плана выполнения программы - T_{plan} ;
- время реализации плана - TE ;

Время реализации плана TE может быть выражено следующей формулой:

$$TE = \sum_{n=1}^k (T_n^R + T_n^T) + T^M + T^{PL}, \text{ где}$$

k - число информационных ресурсов, T_n^R - время выполнения запроса на n -ом ресурсе, T_n^T - время передачи данных от n -ого ресурса в посредник или другой ресурс, T^M - время выполнения запроса в посреднике, T^{PL} - время выполнения программы в языке программирования. Важно отметить что в случае если происходит многократное выполнение одного и того же рассредоточения, то план строится единожды.

Определение 11. Оценкой эффективности рассредоточения будем называть функционал ET (Execution Time), существенно зависящий от рассредоточения, такой что $ET = T_{plan} + TE$.

Определение 12. Минимальным рассредоточением называется такое рассредоточение, при котором оценка эффективности рассредоточения ET будет минимальной среди всех возможных рассредоточений.

Таким образом, постановка задачи данной работы - поиск минимального рассредоточения для

многоязычной спецификации реализации алгоритма решения задачи в среде предметных посредников.

Далее рассмотрен процесс построения рассредоточения, который был применен к конкретной реализации предметных посредников, разрабатываемой в рамках проекта СИНТЕЗ в ИПИ РАН.

5. Описание процесса построения рассредоточения

Оценочная модель для задачи рассредоточения приведена в определении 11. Стоит отметить, что оценивается реальное время выполнения, включающее все накладные расходы. При работе с многоязычными программами [10] важно определить общие структуры, в которые эти программы будут отображены. В модели рассредоточения спецификация алгоритма решения задачи представлена в подобных общих структурах, называемых в работе функциональными операциями. При построении экспертных правил необходимо определить условия их применения. В качестве подобных правил может выступать [8] анализ состояния графа (наличие каких либо шаблонов), либо состояние вершин графа. В модели рассредоточения в качестве вершин графа выступают функциональные операции, поэтому важно определить классы функциональных операций, для которых справедливы общие экспертные правила. Кроме того, определение классов функциональных операций позволяет также ограничить пространство поиска при полном переборе, т.к. не все операции выразимы в каждом из языков спецификации реализации алгоритма решения задачи.

В данной работе исследуются задача рассредоточения спецификации реализации алгоритма решения задачи, заданная в виде многоязычной программы, представленной на языке программирования, языке правил предметных посредников, а также на языке спецификации взглядов. В работе рассматриваются перестановки операций из взглядов в ЯП, из языка правил в ЯП, а также из языка правил во взгляды и обратно. Перестановка операций из ЯП считается невозможной. Поэтому при выделении классов функциональных операций рассматриваются: язык спецификации правил предметных посредников и язык спецификации взглядов.

Таким образом, для среды посредников были выделены 14 классов функциональных операций FC (соответствующие функциональным возможностям языков).

- fc₁ - класс операций предикатов коллекций,
- fc₂ - класс операций отрицания предиката коллекций,
- fc₃ - класс операций отрицания предиката условий,
- fc₄ - класс операций конъюнкции формул,
- fc₅ - класс операций дизъюнкции формул,

fc₆¹ - класс операций функциональных предикатов реализованных информационными ресурсами,

fc₆² - класс операций функциональных предикатов реализованных веб-сервисами,

fc₆³ - класс операций функциональных предикатов реализованных на ЯП,

fc₆⁴ - класс операций функциональных предикатов реализованных на языке программирования объектно-реляционной СУБД предметного посредника,

fc₇ - класс операций предикатов условий,

fc₈¹ - класс операций функциональных предикатов разрешения конфликтов, задаваемых во взглядах, реализованных на ЯП,

fc₈² - класс операций функциональных предикатов разрешения конфликтов, задаваемых на языке программирования объектно-реляционной СУБД предметного посредника,

fc₉ - класс операций предикатов условий над информационными ресурсами,

fc₁₀ - класс операций описываемых в ЯП.

Неоднородность выделенных классов возникает из-за естественной неоднородности компонентов, между которыми осуществляется рассредоточение. Каждый компонент позволяет задавать лишь часть из выше описанных классов операций. Корректными назначениями для классов функциональных операций, будем считать следующие назначения:

назначение программе в СП:

{fc₁, fc₃, fc₄, fc₅, fc₆², fc₆³, fc₇, fc₈¹, fc₁₀}

назначение программе на правилах:

{fc₁, fc₂, fc₃, fc₄, fc₅, fc₆¹, fc₆², fc₆³, fc₆⁴, fc₇, fc₈², fc₉}

назначение взглядам:

{fc₁, fc₂, fc₃, fc₄, fc₇, fc₈¹, fc₈², fc₉}

Так, например, операция дизъюнкции формул может быть описана как на правилах, так и программой в СП.

Максимальная оценка общего числа рассредоточений (если перебирать и все возможные планы выполнения) определяется как $3^n * y^m$. Где 3^n – общее число вариантов рассредоточений, а y^m – общее число планов, для конкретного рассредоточения. Полный перебор не представляется возможным для задач, где требуется многократное исполнение. Поэтому для выполнимых операций вместо полного перебора (y^m) планировщик строит всегда выполнимый квази-оптимальный план. Алгоритм планирования основывается на эвристиках, статистике и оценочных запросах для различных планов, симуляции на неполных выборках (1-5% данных), а также на основании учета возможностей ресурсов (capabilities). Подробно алгоритм планирования описан в работе [2]. Считается что план, построенный планировщиком, не требует улучшения. Таким образом, общее число вариантов рассредоточения сокращается до 3^n .

Алгоритм, использованный при планировании, невозможно применить для функциональных операций и выбора эффективного

рассредоточения вследствие естественной разнородности функциональных компонентов. Например, в среде посредников с компонентами (СП, посредник, адаптеры, ресурсы) планировщик взаимодействует через унифицированный интерфейс, отправляя оценочные запросы и получая статистику. В случае же функциональных компонентов в среде посредников подобная унификация невозможна, в силу того что рассредоточение осуществляется на уровне исходных языков. Изначально происходит сопряжение разнородных компонентов: ЯП, языка спецификации правил, язык спецификации взглядов, спецификаций адаптеров и спецификаций ресурсов. Если бы спецификация реализации задавалась на одном языке, задача была бы совсем иной.

Для построения эффективного рассредоточения были разработаны следующие методы:

- метод построения модели рассредоточения;
- методы перестановки операций в модели рассредоточения;
- метод прогонки.

Метод построения модели рассредоточения включает в себя:

- построение графа зависимости функциональных операций,
- определение назначения для всех операций на основании начальной реализации,
- определение классов всех операций модели рассредоточения,
- определение возможных назначений для всех операций.

Метод прогонки – это выполнение всей реализации задачи с ограниченной выборкой используемых данных в ресурсах. Таким образом, можно существенно сокращать время оценки конкретного рассредоточения.

Сама по себе задача перестановки операций нетривиальна в силу естественной разнородности компонентов назначений и языков, на которых задаются операции (язык программирования, язык правил, язык спецификации взглядов). Для перестановки операций используются два метода. Метод основанный на эвристических правилах, и метод, основанный на оценке производительности.

Метод перестановки операций, основанный на оценке производительности, заключается в полном переборе возможных перестановок операций, и оценки производительности методом прогонки. Как было сказано выше, спецификация реализации отображается в модель рассредоточения. Каждая операция характеризуется своим классом (классы функциональных операций описаны выше), а также назначением. Для осуществления полного перебора необходимо определить все возможные варианты перебора, для этого используется следующий алгоритм.

В начале, строятся все возможные варианты назначений. Таких вариантов – 3^n . Затем среди всех

вариантов отсеиваются те, которые не удовлетворяют корректности возможных назначений, определенной выше. Например, операции класса fc_5 не могут быть специфицированы на языке спецификации взглядов. Наконец, среди оставшихся отсеиваются те варианты, которые не удовлетворяют зависимостям операций в графе зависимостей.

Когда все корректные варианты построены, для каждого выполняется прогон, после чего выбирается минимальное рассредоточение.

Метод перестановки операций на основании эвристических правил заключается в применении экспертных правил для анализа возможности перестановки каждой из операций. Метод основывается на направленном поиске эффективного рассредоточения в графе зависимостей функциональных операций. В графе отсутствуют циклы (т.к. операции не могут быть взаимозависимыми). Метод полуавтоматический, рассчитан на взаимодействие с экспертом.

Последовательно применяются все эвристические правила, пока они изменяют модель рассредоточения, улучшая ее. После чего эксперту предлагается переназначить те операции, для которых вовремя прогонки определено, что их выполнение не эффективно. После чего экспертные правила могут быть снова применены.

Эксперименты показали применимость подхода по построению рассредоточения. В случае поиска минимального рассредоточения полным перебором выявлены следующие свойства подхода:

- разрешимость - минимальное рассредоточение поддается оценке при любом начальном рассредоточении,
- устойчивость – минимальное рассредоточение не зависит от объема данных в ресурсах в случае достаточно больших объемов данных,
- масштабируемость – оценка эффективности растет линейно с ростом числа ресурсов и числа классов посредников.

Стоит отдельно отметить, что описанные выше свойства получены на примере решения двух задач, поэтому не могут обладать абсолютной общностью. При тестировании подхода построения рассредоточений на основе экспертных правил, эксперименты показали приемлемые результаты, если для функций определены различные реализации (веб-сервисы, на ЯП, на языке программирования СУБД). В этом случае на основании оценок можно было выбрать лучшую, с точки зрения эффективности, реализацию. В противном случае, автоматически построить хорошее рассредоточение значительно сложнее. Вместе с тем, экспертом может быть получено приемлемое рассредоточение после применения экспертных правил.

6. Заключение

В работе приведено описание подхода построения минимального рассредоточения для многоязычной спецификации алгоритма решения задачи. В работе дается определение рассредоточения, оценки эффективности рассредоточения, а также минимального рассредоточения. В работе рассмотрена модель рассредоточения, а также выделены классы функциональных операций, являющихся общими для различных языков. Наконец, в работе приводится описание методов построения рассредоточения.

Литература

- [1] Брюхов Д.О. Вовченко А.Е. Захаров В.Н. Желенкова О.П. Калинин Л.А. Мартынов Д.О. Скворцов Н.А. Ступников С.А. Архитектура промежуточного слоя предметных посредников для решения задач над множеством интегрируемых неоднородных распределенных информационных ресурсов в гибридной грид-инфраструктуре виртуальных обсерваторий. Информатика и ее применения. – 2008. — Т. 2, вып. 1. – С. 2 – 34
- [2] Вовченко А.Е., Крупа А.В. Планирование запросов над множеством неоднородных распределенных информационных ресурсов в архитектуре средств поддержки предметных посредников. Труды 11ой Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» - RCDL'2009, Петрозаводск, Россия, 2009. С. 335-342.
- [3] S. Adali, K. S. Candan, Y. Papakonstantinou, V. S. Subrahmanian. Query caching and optimization in distributed mediator systems. Proceedings of the 1996 ACM SIGMOD international conference on Management of data. New York, NY, USA, 1996.
- [4] José Luis Ambite, Craig A. Knoblock. Flexible and scalable cost-based query planning in mediators: a transformational approach. Journal: Artificial Intelligence - Special issue on Intelligent internet systems. Volume 118 Issue 1i2, April 2000.
- [5] J.L. Ambite, C.A. Knoblock, Planning by rewriting: Efficiently generating high-quality plans, in: Proc. AAAI-97, Providence, RI, 1997.
- [6] Briukhov D.O., Kalinichenko L.A., Martynov D.O. Source Registration and Query Rewriting Applying LAV/GLAV Techniques in a Typed Subject Mediator. Proc. of the Ninth Russian Conference on Digital Libraries RCDL'2007.
- [7] Surajit Chaudhuri. An Overview of Query Optimization in Relational Systems. Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems. 1998.
- [8] Rajiv Dewan, Abraham Seidmann, Zhiping Walter. Workflow Optimization through Task Redesign in Business Information Processes. Proceeding HICSS '98 Proceedings of the Thirty-First Annual Hawaii International Conference on System Sciences - Volume 1, IEEE Computer Society Washington, DC, USA, 1998.
- [9] J. Ebert, A. Winter, P. Dahm, A. Franzke, and R. Süttenbach. Graph Based Modeling and Implementation with EER/GRAL. In B. Thalheim, editor, 15th International Conference on Conceptual Modeling (ER'96), Proceedings, number 1157 in LNCS, pages 163–178. Springer, Berlin, 1996.
- [10] Bernt Kullbach, Andreas Winter, Peter Dahm, Jurgen Ebert. Program Comprehension in Multi-Language Systems. Proceeding WCRE '98 Proceedings of the Working Conference on Reverse Engineering (WCRE'98), 1988.
- [11] J. Sowa. Conceptual Structures. Information, Processing in Mind and Machine. The Systems Programming Series. Addison-Wesley, Reading, 1984.
- [12] Ramana Yerneni, Chen Li, Jeffrey D. Ullman, Hector Garcia-Molina. Optimizing Large Join Queries in Mediation Systems. Proceeding ICDT '99 Proceedings of the 7th International Conference on Database Theory, 1999.

Dispersed Organization of Problem Solving in the Distributed Environment of Subject Mediators

© A.E. Vovchenko

Methods for dispersed organization of problem solving in the mediation environment are discussed in the paper. The problem arises from the fact that the parts of a problem solving algorithm may be assigned to various components of the mediation environment and specified in different languages, e.g., in programming language, in rule-based language, in the form of views. The paper presents a model of dispersed organization for multi-language programs, as well as an approach for constructing minimal dispersed implementation.

* Работа выполнена при частичной финансовой поддержке РФФИ (гранты 10-07-00342-а, 11-07-00402-а, 10-07-00640-а) и Программы фундаментальных исследований Президиума РАН №15, проект 4.2.