

## РАССРЕДОТОЧЕНИЕ РЕАЛИЗАЦИИ АЛГОРИТМА РЕШЕНИЯ ЗАДАЧ НАД НЕОДНОРОДНЫМИ РАСПРЕДЕЛЕННЫМИ РЕСУРСАМИ

А. Е. Вовченко

*ИПИ РАН*

Поступила в редакцию 20.10.2011 г.

**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 оптимального рассредоточения.

**Ключевые слова:** предметные посредники, оптимизация, неоднородные распределенные ресурсы, решение научных задач, планирование, интеграция

**Annotation.** 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.

**Key words:** subject mediators, optimization, heterogeneous distributed resources, science problem solving, planning, integration

### ВВЕДЕНИЕ

В различных областях науки наблюдается экспоненциальный рост объема получаемых экспериментальных (наблюдательных) данных. 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.

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.

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 языке программирования (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 посредника на языке правил; GLAV [5] взгляды, определяемые при регистрации ресурсов; программируемые

адаптеры ресурсов данных и сервисов; информационные ресурсы.

В следующем разделе представлена обобщенная архитектура среды предметных посредников. В разделе 3 приведен обзор существующих подходов. Далее рассматривается определение рассредоточения в среде предметных посредников, а также функций оценки рассредоточения. Затем представлен подход к построению оптимального рассредоточения в среде посредников.

### ОБОБЩЕННАЯ АРХИТЕКТУРА СРЕДЫ ПРЕДМЕТНЫХ ПОСРЕДНИКОВ

Обобщенная архитектура среды предметных посредников, представленная на рисунке 1, включает:

- уровень информационных ресурсов, включающий базы данных, сервисы, средства программирования, а также другие средства, возможно организованные в рамках грид, облачных или других инфраструктур;
- слой адаптеров, обеспечивающий interoperability ресурсов благодаря технической унификации их интерфейсов и введению дистанционных механизмов обращения к ресурсам;

- 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, 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.

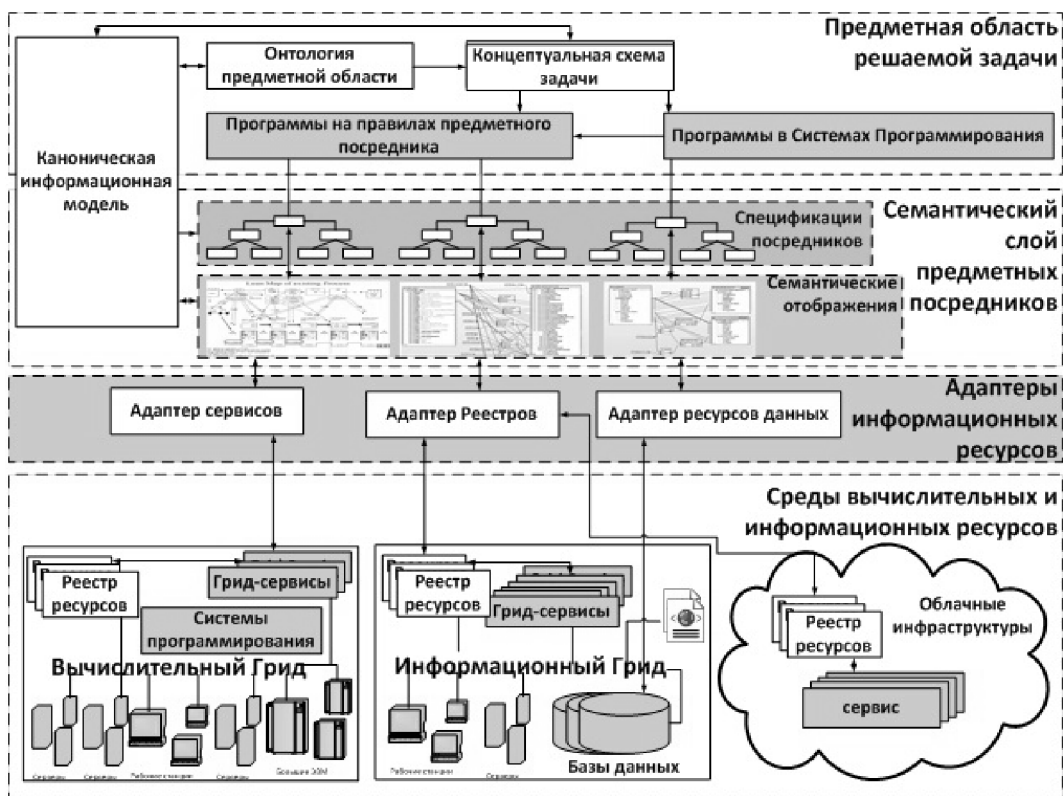


Рис. 1. Обобщенная архитектура среды предметных посредников

## 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. Так, задача близка к подходам построения эффективных планов выполнения запросов в среде предметных посредников [4, 6] и к проблеме выбора оценочной модели, а также вопросам кэширования [3] при оптимизации запросов. Задача также близка к работам, посвященным анализу многоязычных программ [9] (программ заданных на нескольких языках программирования). Наиболее распространенный подход для анализа программ на разных языках – это отображение программы во множество общих конструкторов, в рамках которых уже осуществляется анализ. В различных проектах в качестве общих конструкторов используются ERM-модели, либо много чаще используются концептуальные графы [8]. Наконец задача построения рассредоточений близка к работам по оптимизации потоков работ [7], представленным графами, т.к. минимальное рассредоточение строится путем перестановки операций в графе (модели рассредоточения).

### ПОСТАНОВКА ЗАДАЧИ РАССРЕДОТОЧЕНИЯ В СРЕДЕ ПРЕДМЕТНЫХ ПОСРЕДНИКОВ

Реализация задается в виде программ и спецификаций следующих компонентов среды:

- языков и систем 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: предметных посредников, программ, порожденных системами программирования, взглядов, адаптеров, ресурсов.

Реализация алгоритма решения задачи может быть задана не единственным образом.

Потенциал каждого из компонентов ограничен его функциональными возможностями.

**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  $op2$ .

**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 именуются 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  $op2$ .

**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.

**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, выполняе-

мые 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, зависящих 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. При рассмотрении рассредоточения между тремя компонентами – взглядами, правилами, ЯП, общее число вариантов рассредоточения для функциональных операций –  $3^n$ .

План выполнения строится исходя из рассредоточения. В рассредоточении выбираются те операции, для которых назначения – взгляды, по множеству этих операций восстанавливается 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, –  $u^m$ , где  $u$  – количество компонентов, на которых происходит выполнение (учитываются все адаптеры, посредник, интерпретаторы систем программирования),  $m$  – количество исполняемых операций.

Эффективность рассредоточения оценивается временем, затрачиваемым на решение 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:

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

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

**Определение 10.** Минимальным рассредоточением называется такое рассредоточение, при котором оценка эффективности рассредоточения  $ET$  будет минимальной среди всех возможных рассредоточений.

## ОПИСАНИЕ ПРОЦЕССА ПОСТРОЕНИЯ РАССРЕДОТОЧЕНИЯ

Оценочная модель для задачи рассредоточения приведена в определении 9. Стоит отметить, что оценивается реальное время выполнения, включающее все накладные расходы. При работе с многоязычными программами [9] важно определить общие конструкты, в которые эти программы будут отображены. В работе подобные общие конструкты названы функциональными операциями. 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.

Для среды посредников были выделены 7 классов функциональных операций  $FC$  (соответствующие функциональным возможностям языков).

$fc_1$  – класс операций предикатов коллекций,



$fc_2$  – класс операций отрицания предиката коллекций,

$fc_3$  – класс операций отрицания предиката условий,

$fc_4$  – класс операций конъюнкции формул,

$fc_5$  – класс операций дизъюнкции формул,

$fc_6$  – класс операций функциональных предикатов,

$fc_7$  – класс операций предикатов условий,

Максимальная оценка общего числа расщедоточений (если перебирать и все возможные планы выполнения) определяется как  $Z^n * y^m$ . Где  $Z^n$  – общее число вариантов расщедоточений, а  $y^m$  – общее число планов, для конкретного расщедоточения. Полный перебор не представляется возможным для задач, где требуется многократное исполнение. Поэтому для выполнимых операций вместо полного перебора ( $y^m$ ) планировщик строит всегда выполнимый 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 способностей ресурсов (capabilities). Подробно алгоритм планирования описан в работе [2]. Считается что план, построенный планировщиком, не требует улучшения. Таким образом, общее число вариантов расщедоточения сокращается до  $Z^n$ .

Для построения эффективного расщедоточения были разработаны следующие методы:

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

Метод построения модели расщедоточения включает:

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

Для перестановки операций используются два метода: метод, основанный на экспертных правилах, и метод, основанный на оценке производительности.

Метод перестановки операций, основанный на оценке производительности, заключается в направленном переборе возможных перестановок операций и оценке производительности методом прогонки. Для осуществления полного перебора необходимо определить всевозможные варианты перебора, для этого используется следующий алгоритм.

Вначале, строятся всевозможные варианты расщедоточений. Таких вариантов –  $Z^n$ . Затем среди всех вариантов отсеиваются те, которые не удовлетворяют корректности возможных назначений. 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 расщедоточение. Также возможно применение направленного перебора, когда операции рассматриваются независимо. В этом случае число вариантов снижается до  $Z * n$ .

Метод перестановки операций на основании экспертных правил заключается в применении экспертных правил для анализа возможности перестановки каждой из операций. Метод полуавтоматический, рассчитан на взаимодействие с экспертом. Последовательно применяются все экспертные правила, пока они изменяют модель расщедоточения, 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 расщедоточения. В случае поиска минимального расщедоточения полным перебором выявлены следующие свойства подхода:

- разрешимость – минимальное расщедоточение поддается оценке при любом начальном расщедоточении,
- устойчивость – минимальное расщедоточение 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.

## ЗАКЛЮЧЕНИЕ

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

## СПИСОК ЛИТЕРАТУРЫ

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 тех-

нологии, 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. Josü 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. 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.

6. 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.

7. 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.

8. J. Ebert, A. Winter, P. Dahm, A. Franzke, and R. Söttenbach. 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.

9. Berndt Kullbach, Andreas Winter, Peter Dahm, Jürgen Ebert. Program Comprehension in Multi-Language Systems. Proceeding WCRE '98 Proceedings of the Working Conference on Reverse Engineering (WCRE'98), 1988.

**Вовченко А. Е.** – м.н.с., ИПИ РАН, E-mail: itsnein@gmail.com.

**Vovchenko A. E.** – IPI RAN, Junior Researcher, E-mail: itsnein@gmail.com.