

# Полуавтоматическое GLAV-отображение спецификаций в канонической информационной модели предметных посредников\*

© Рябухин О.В.

ВМК МГУ, Институт проблем информатики РАН  
ovr@ipi.ac.ru

## Аннотация

Предметный посредник представляет собой компонент промежуточного слоя, он занимает место между ресурсами и пользователем (приложением) и предоставляет унифицированный доступ к множеству зарегистрированных в нем ресурсов. Предметный посредник строится для решения класса задач конкретной предметной области (в таких научных областях, как, например, в области наук о Земле, астрономии, биологии, и других). Данная работа посвящена полуавтоматическому построению GLAV (Global-Local-As-View)-отображений между спецификациями информационных ресурсов и посредника. В работе сформулирована задача отображения спецификаций как часть процесса регистрации ресурса в посреднике и описана идея ее полуавтоматического решения; приведен обзор существующих решений; представлен прототип системы поддержки процесса регистрации, реализующий описанный подход к полуавтоматическому построению отображений спецификаций.

компонентом промежуточного слоя и занимает место между ресурсами и пользователем (приложением), предоставляя унифицированный доступ к множеству зарегистрированных в нем ресурсов. Предметный посредник строится для решения класса задач конкретной предметной области (в таких научных областях, как, например, в области наук о Земле, астрономии, биологии, и других). Посредник определяется своей концептуальной моделью. В нее входит концептуальная схема посредника (включающая спецификации абстрактных типов данных, классов, атрибутов, ассоциаций, функций), а также онтология предметной области.

В данной работе рассматриваются виртуальные посредники, движимые приложением (или классом задач). В виртуальном посреднике не происходит материализации данных реальных ресурсов. Виртуальный посредник связывается с ресурсами с помощью декларативных правил, определяющих, каким образом информация в ресурсе должна соответствовать информации, предоставляемой посредником. Запрос к посреднику с помощью указанных правил переписывается в подзапросы к ресурсам (на соответствующих им языках), после чего они исполняются и объединенный результат возвращается пользователю. Виртуальная интеграция имеет преимущества перед интеграцией с материализацией данных, т.к. не отстает от состояния ресурсов. Посредники, движимые приложением, строятся независимо от существующих информационных ресурсов. Это позволяет создавать системы, масштабируемые по числу ресурсов. Подход, движимый ресурсами, таким достоинством не обладает, т.к. при появлении нового ресурса, с большой вероятностью схема посредника должна быть модифицирована.

## 1 Введение

### 1.1 Предметные посредники

В настоящее время наблюдается значительный рост количества разнообразной информации, накапливаемой в информационных ресурсах (базах данных, сервисах, потоках работ). Решение задач в таких условиях требует совместного использования информации в ресурсах, ее интеграции. Среда посредников предоставляет возможность решения задач над множеством неоднородных информационных ресурсов. Посредник является

### 1.2 Контекст исследований

Данная работа является частью проекта SYNTHESIS [11] (СИНТЕЗ (Система ИНТЕграции Знаний)), исследования которого, в частности, направлены на применение подхода виртуальных посредников, движимых приложениями, для решения задач над множеством неоднородных распределенных ресурсов. При этом рассматриваются разнообразные ресурсы: базы

данных различных моделей данных, сервисы, потоки работ. Для того чтобы иметь возможность построить посредник, позволяющий работать с таким многообразием информационных моделей (реляционных, объектных, сервисных моделей, моделей потоков работ), была разработана каноническая объектно-фреймовая информационная модель, определяемая языком СИНТЕЗ [1]. Единицей описания в языке СИНТЕЗ является фрейм. На базе языка фреймов создана объектная модель языка. В основе объектной модели лежит понятие абстрактного типа данных, служащее для описания типов данных любой природы. Его описание включает спецификации атрибутов, инвариантов и функций. Иерархия типов задается отношением подтипа. В СИНТЕЗе определен набор базовых типов. Классы в языке СИНТЕЗ представляют совокупности однородных объектов. Каждый объект из совокупности является экземпляром данного класса. Классы принадлежат иерархии, основанной на отношении обобщения/специализации.

Спецификации ресурсов и посредника представляют собой совокупность их типов, классов и функций. В проекте СИНТЕЗ применяется техника соотнесения спецификаций ресурсов со спецификациями посредников на основе взглядов [6] – спецификации связываются правилами, выражающими взгляды (views). Посредникам, движимым приложениями, соответствует техника LAV (Local-as-View), при которой классы ресурса выражаются при помощи классов посредника и их композиций. Ее противоположностью является техника GAV (Global-As-View), при которой классы посредника выражаются при помощи композиций классов ресурсов. В проекте СИНТЕЗ используется комбинированный подход, объединяющий преимущества упомянутых техник, именуемый GLAV (Global-Local-As-View). В этом случае преобладающей является техника LAV, однако выражения взглядов могут содержать конструкции из классов ресурсов (что соответствует GAV), а также дополнительные функции, с помощью которых возможно разрешение структурных конфликтов.

Выражения взглядов описываются с помощью языка формул СИНТЕЗа, который является вариантом типизированного языка логики первого порядка. Предикаты в формулах соответствуют классам и функциям. Правила (взгляды) GLAV имеют вид:

$$Cr_1(vr_1/T_{vr_1}), \dots, Cr_n(vr_n/T_{vr_n}), F_1(t_1, y_1), \dots, F_m(t_m, y_m), Br, - \\ Cm_1(vm_1/T_{vm_1}), \dots, Cm_n(vm_n/T_{vm_n}), Vm$$

где  $Cr_i$  ( $Cm_i$ ) – предикат класса ресурса (посредника),  $F_i$  – функциональный предикат,  $Br$ ,  $Vm$  – ограничения (условия). Переменные  $vr_1, \dots, vr_n, vm_1, \dots, vm_n$ , типизированы  $T_{vr_1}, \dots, T_{vr_n}, T_{vm_1}, \dots, T_{vm_n}$  соответственно.

## 2 Задача отображения спецификаций

Как отмечалось выше, посредник строится независимо от ресурсов. Для того чтобы наполнить его информацией, ресурсы необходимо зарегистрировать в нем. Основной операцией при регистрации является отображение спецификаций ресурса в спецификации посредника (предполагается, что спецификации уже выражены в канонической информационной модели).

**Определение 1.** Отображением спецификаций в канонической информационной модели называется тройка  $(R, M, V)$ , где  $R$  – спецификации ресурса, выраженные в канонической модели,  $M$  – спецификации посредника,  $V$  – множество взглядов GLAV, связывающих классы спецификаций  $R$  и  $M$ .

Одной из отличительных черт СИНТЕЗа является подход к регистрации, при котором необходимо, чтобы регистрируемые спецификации ресурса уточняли спецификации посредника. Говоря неформально, можно утверждать, что спецификация  $A$  уточняет спецификацию  $B$ , если пользователь не замечает подмены при использовании  $A$  вместо  $B$ . Т.к. не каждая спецификация ресурса уточняет спецификацию посредника, в процессе регистрации необходимо произвести поиск фрагментов отображаемых спецификаций, находящихся в отношении уточнения (в эти фрагменты могут входить как типы, классы и функции, так и их композиции).

**Определение 2.** Спецификация  $S_1$  называется фрагментом спецификации  $S_2$ , если:

- множество классов  $S_1$  является подмножеством множества классов  $S_2$  и/или их композиций
- множество функций  $S_1$  является подмножеством множества функций  $S_2$
- множество типов (или редуктов типов)  $S_1$  является подмножеством типов (или редуктов типов)  $S_2$  и/или их композиций.

**Определение 3.** Задачей отображения спецификаций  $R$  (ресурса) и  $M$  (посредника) в канонической информационной модели называется задача поиска отображения  $(S_r, S_m, V)$ , такого, что:

- $S_r, S_m$  – фрагменты спецификаций  $R$  и  $M$  соответственно
- типы и функции  $S_r$  уточняют типы и функции  $S_m$
- ограничения целостности, определенные в  $S_r$ , удовлетворяют ограничениям целостности, определенным в  $S_m$  (в ограничения целостности входят инварианты классов и утверждения (assertions), вынесенные за рамки типов и классов)

Композиционное исчисление спецификаций [2] используется для формализации процесса композиции и декомпозиции их фрагментов. Важнейшими здесь являются понятия редукта типа и наибольшего общего редукта двух типов, а также операций композиции типов.

**Определение 4.** Редукт  $R_T$  типа  $T$  определяется как подсигнатура сигнатуры типа, при этом множество атрибутов редукта является

подмножеством множества атрибутов типа, множество функций редукта является подмножеством множества функций типа, множество предикатов редукта является подмножеством множества предикатов типа.

**Определение 5.** Наибольший общий редукт  $R_{MC}(T_1, T_2)$  для типов  $T_1, T_2$  есть редукт  $R_{T_1}$  типа  $T_1$  такой, что существует редукт  $R_{T_2}$  типа  $T_2$  который является уточнением редукта  $R_{T_1}$  и не существует другого редукта  $R'_{T_1}$  такого, что  $R_{MC}(T_1, T_2)$  является редуктом  $R'_{T_1}$ ,  $R'_{T_1}$  не равен  $R_{MC}(T_1, T_2)$  и существует  $R'_{T_2}$  являющийся уточнением редукта  $R'_{T_1}$ . Редукт  $R_{T_2}$  называется сопряженным по отношению к  $R_{T_1}$ .

**Определение 6.** Операция meet. Операция образует тип  $T$  как пересечение спецификаций типов-операндов. Тип  $T$  образуется слиянием двух наибольших общих редуктов типов  $T_1$  и  $T_2$ :  $R_{MC}(T_1, T_2)$  и  $R_{MC}(T_2, T_1)$ . Слияние двух редуктов включает в себя объединение множеств их функций. Если в объединении имеется две функции, находящиеся в отношении уточнения, то только одна из них – более абстрактная – включается в результирующий тип  $T$ . Инвариант результирующего типа формируется как дизъюнкция инвариантов, взятых из спецификаций наибольших общих редуктов.

**Определение 7.** Операция join. Операция образует тип  $T$  как объединение спецификаций типов-операндов. Тип  $T$  образуется слиянием спецификаций типов  $T_1$  и  $T_2$ . Общие элементы спецификаций типов  $T_1$  и  $T_2$  включаются в результирующий тип только один раз. Общие элементы определяются посредством слияния сопряженных редуктов двух наибольших общих редуктов типов  $T_1$  и  $T_2$ :  $R_{MC}(T_1, T_2)$  и  $R_{MC}(T_2, T_1)$ . Слияние сопряженных редуктов включает в себя объединение множеств их функций. Если в объединении имеется две функции, находящиеся в отношении уточнения, то только одна из них – более точная – включается в результирующий тип  $T$ . Инвариант результирующего типа формируется как конъюнкция инвариантов, взятых из спецификаций типов операндов  $T_1$  и  $T_2$ .

Таким образом, подход СИНТЕЗа к отображению спецификаций [3] состоит из следующих шагов:

1. Выделение фрагментов отображаемых спецификаций, находящихся в отношении уточнения, при помощи операции взятия наибольшего общего редукта.

2. Формирование композиций фрагментов спецификаций, находящихся в отношении уточнения (с помощью операций композиции типов, применяемых к редуктам, построенным на предыдущем шаге).

3. Формирование взглядов GLAV.

### 3 Полуавтоматическое отображение спецификаций

Построение отображения спецификаций вручную – трудоемкий процесс, поэтому необходимы программные средства его поддержки. Идея автоматизации построения отображений в подобной системе такова: между спецификациями ресурса и посредника необходимо найти элементарные соответствия (например, некий атрибут ресурса соответствует атрибуту посредника), исходя из которых возможно построить наибольшие общие редукты, по которым сформировать взгляды.

Таким образом, полуавтоматическое построение отображения происходит согласно следующему процессу:

```

Input = (R, M)
//R - спецификации ресурса, M - спецификации посредника

Match(R, M) = { (a, b) | a ∈ R, b ∈ M, a ~ b }
//сопоставление спецификаций и формирование множества пар релевантных (a~b) элементов
Match(R, M)

MCReducts(R, M) = MCReducts(R, M, Match(R, M)) = {
Rmc(t1, t2) | t1 ∈ ADT(R), t2 ∈ ADT(M), t1~t2}
//по результатам Match(R, M) формируется множество наибольших общих редуктов
MCReducts(R, M)

V = Views(R, M, MCReducts(R, M)) = {v | v - GLAV взгляд}
//по результатам конструирования наибольших общих редуктов формируется набор GLAV-взглядов

Result = (R, M, V)
//искомое отображение

```

Подход является полуавтоматическим, т.к. на каждом шаге эксперт должен подтвердить, либо редактировать предложенное системой решение. Для сопоставления спецификаций применяются техники сопоставления схем (schema matching [8]), которые используют эвристики (например, сравнение имен или описаний элементов) для определения похожих частей спецификаций. Алгоритмизация операций взятия наибольшего общего редукта и композиций типов приведена в работе [5]. Теоретическая часть данной работы рассматривает заключительный шаг: формирование взглядов.

### 4 Формирование выражений взглядов

Необходимо развитие известных алгоритмов формирования выражений взглядов (например, [4]) в следующих направлениях:

- формирование выражений взглядов в случае сложных связей между классами посредника в правилах (например, если класс ресурса выражается при помощи конъюнкции нескольких классов посредника при некотором условии)

- формирование условий во взглядах (построение условий может быть автоматизировано путем анализа ограничений целостности)

#### 4.1 Формирование взглядов при наличии сложных связей между классами

В случае, если один класс ресурса выражается при помощи композиции нескольких классов посредника, необходимо определить, для каких классов из них требуется задать во взгляде конъюнкцию, а для каких - объединение. Для каждой из объединяемых частей формулы формируется отдельный взгляд (формирование такого взгляда соответствует операции объединения, т.к. все множество взглядов (правил) интерпретируется как их объединение (семантика UCQ)). Конъюнкции классов интерпретируются как соединение, если у типов их экземпляров есть общий редукт, или как декартово произведение при отсутствии общего редукта.

В качестве примера рассмотрим следующую ситуацию: пусть посредник предоставляет интегрированную информацию о музейных экспонатах. В посреднике определены классы `creator` (хранит данные о создателях экспонатов: имена и дополнительную информацию) и `painting` (данные о картинах: названия, имена художников и др.):

```
{Creator; in: type;
  name: string;
  culture_race: string;
},
{Painting; in: type;
  title: string;
  created_by: string;
  place_of_origin: string;
  date_of_origin: date;
  general_info: string;
}
```

Также имеется ресурс, который содержит информацию о полотнах некоторого музея (класс `canvas` – данные о полотнах: название, имя художника, описание и др.):

```
{Canvas; in: type;
  title: string;
  painter: string;
  culture: string;
  date: date;
  place: string;
  description: string;
}
```

В данном случае класс ресурса `canvas` соответствует соединению классов посредника `creator` и `painting` по условию `creator.name = painting.created_by`. Т.е. `canvas` выражается следующим образом:

```
canvas(x[...]) :- creator(y/[name, ...] &
  painting(z/[created_by, ...] & y.name = z.created_by
```

Одним из способов автоматизации формирования взглядов в таком случае является анализ ограничений целостности,

сформулированных в посреднике или ресурсе [7] (например, внешних ключей). Этот подход позволяет определить как вид операций (соединение, объединение), необходимых для композиции классов, так и определить наибольший общий редукт в случае, если необходимо соединение. Например, если между двумя классами определено ограничение внешнего ключа, то это может служить основанием для их соединения по соответствующим атрибутам: например, если в рассматриваемом примере в посреднике имеется ограничение внешнего ключа вида:

```
all x/Painting(ex y/Creator (x.created_by =
  y.name)) (для каждой картины также имеется
  информация о художнике, нарисовавшем ее)
```

Мы можем сделать вывод о том, что классы `painting` и `creator` можно соединить по условию `painting.created_by = creator.name`.

В качестве алгоритма для анализа ограничений целостности можно использовать адаптацию алгоритма прогонки (`chase[9]`): рассматривая последовательно ограничения целостности, мы можем определить группы логически связанных классов.

Если из анализа ограничений целостности необходимых выводов сделать не удалось, эксперт может самостоятельно указать вид операций, необходимых для композиции классов. Далее система строит наибольшие общие редукты типов экземпляров классов, для которых требуется соединение.

Также, система может хранить результаты регистрации ресурсов и при регистрации нового ресурса предлагать для композиции классов посредника те операции, которые использовались ранее.

#### 4.2 Автоматизация формирования условий во взглядах

Формирование условий во взгляде может быть частично автоматизировано путем анализа ограничений целостности. В качестве примера рассмотрим следующую ситуацию: пусть в некоторой компании имеется посредник, предоставляющий интегрированную информацию о ее сотрудниках из разных филиалов. В посреднике присутствует класс `newcomer` с атрибутами `name` и `yearOfEmployment`, который соответствует новичкам, недавно пришедшим в компанию. Тип `Newcomer` является типом его экземпляров и содержит инвариант:

```
all x/Newcomer(x.yearOfEmployment > 2009),
который говорит о том, что новичком считается
сотрудник, пришедший в компании позднее 2009
года. Тогда, если мы регистрируем базу данных
сотрудников филиала, содержащую класс employee
с атрибутами name, yearOfEmployment, который
соответствует всем сотрудникам филиала, то взгляд,
связывающий эти два класса может иметь
следующий вид:
```

*employee(x/[name, yearOfEmployment]) & yearOfEmployment > 2009 :- newcomer(x/[name, yearOfEmployment]).*

Таким образом, ограничения целостности, верные для всех экземпляров класса (т.е. ограничения на переменные, связанные квантором всеобщности), могут быть вынесены во взгляд в качестве условий. Причем, если ограничение целостности сформулировано в посреднике, то оно выносится в голову взгляда, если в ресурсе, то в тело взгляда.

## 5 Обзор существующих решений

В результате анализа работ, близких к проводимым исследованиям, к рассмотрению были выбраны проекты Clio и Automed.

Clio [12] - система интеграции информационных ресурсов, содержащих реляционные или XML-данные. Для отображения схем (реляционных и xml-схем) используется подход GLAV. Clio имеет свой аналог системы поддержки регистрации - систему управления отображением схем (Schema Mapping Management System), в которой возможно сопоставление спецификаций и формирование взглядов. На этапе сопоставления спецификаций выделяются пары релевантных элементов, после чего по полученным соответствиям и ограничениям целостности формируются выражения взглядов.

Automed [13] - инструментарий, в основе которого лежит подход BAV (both as view). Automed способен интегрировать ресурсы широкого класса моделей (реляционные, объектные и др.), что достигается за счет использования модели HDM (Hypergraph Data Model) для описания спецификаций ресурсов. Automed предоставляет систему для сопоставления и слияния схем (Schema Matching And Merging Tool). По результатам сопоставления схем система формирует новую схему, полученную в результате их слияния, а также правила BAV, которые описывают соответствующие трансформации данных.

В рассмотренных решениях (Clio, Automed) алгоритм формирования взглядов принимает на вход результат сопоставления спецификаций, т.е. набор пар элементарных соответствий между их элементами. Таким образом, семантическая правильность отображения определяется правильностью выбора соответствий. Хотя такой подход дает хорошие результаты для таких моделей данных, как реляционная, его оказывается недостаточно в случае объектной модели. Для отображения функций (заданных своими пред- и пост- условиями) необходим формальный аппарат, который предоставляет композиционное исчисление спецификаций. Таким образом, в семантическом плане подход, используемый в СИНТЕЗе, является более богатым, т.к. вносит дополнительный этап - построение наибольших общих редуктов и их композиций, в ходе которого определяются фрагменты спецификаций ресурса,

уточняющих фрагменты спецификаций посредника. При этом, сопоставление спецификаций с помощью эвристики используется лишь как предварительный шаг, помогающий эксперту получить начальное приближение в формировании наибольших общих редуктов. Позднее эксперт может расширить предложенное решение (например, описать дополнительные функции разрешения конфликтов), а также формально доказать факт уточнения спецификаций.

## 6 Система поддержки процесса регистрации

В рамках данной работы была разработана система поддержки процесса регистрации, являющаяся полуавтоматическим инструментарием. Для каждого шага регистрации система предлагает возможные решения, которые эксперт подтверждает, либо редактирует. Для определения спецификаций в системе используется собственный графический редактор, который представляет конструкции языка СИНТЕЗ в графическом виде. При регистрации ресурса система поддерживает следующие этапы:

- сопоставление спецификаций ресурса и посредника
- сопоставление структуры объектных типов ресурса и посредника (поиск релевантных атрибутивных путей)
- конструирование наибольших общих редуктов
- конструирование композиций типов
- формирование выражений взглядов

При сопоставлении спецификаций система сравнивает имена элементов и их текстовые описания на естественном языке.

При сопоставлении структур объектных типов используется понятие релевантного пути [5]. Спецификации пары типов (ресурса и посредника) представляются в особой форме (в виде дерева, которое может быть получено из представления спецификаций типов в виде графа с помощью введения дублирующих вершин). После чего по правилам, сформулированным в работе [5], устанавливается релевантность атрибутивных путей.

В системе реализован алгоритм конструирования наибольших общих редуктов, описанный в работе [5]. Построение искоемых редуктов происходит по результатам сопоставления спецификаций. При наличии структурных конфликтов на этом шаге формируются функции разрешения конфликтов.

В системе возможно конструирование композиционных типов. При этом, эксперт выбирает пару типов, указывает необходимую операцию исчисления спецификаций (соединение или объединение), после чего система строит композиционный тип, исходя из формального определения этих операций [5].

Формирование взглядов происходит в соответствии с вышеописанным алгоритмом.

## 7 Направления дальнейших исследований

### 7.1 Сопоставление спецификаций с использованием онтологий

В настоящее время активно развиваются методы, в основе которых лежит использование онтологий. Применение этих методов при сопоставлении спецификаций может улучшить результат работы системы. Примером является аннотирование элементов концептуальной схемы понятиями онтологии. При наличии таких онтологических аннотаций в спецификациях ресурса и посредника возможно использовать методы онтологической интеграции для определения релевантных элементов [10].

### 7.2 Проверка ограничений целостности

При формировании взглядов необходима проверка того, что ограничения целостности (инварианты классов, утверждения), определенные для отображаемых фрагментов спецификаций ресурса, удовлетворяют соответствующим ограничениям целостности посредника. Такая проверка возможна, например, с помощью алгоритма прогонки (chase [9]). Алгоритм прогонки применим в случае определенного вида ограничений целостности и гарантированно завершается только при определенных условиях. К настоящему времени класс допустимых ограничений довольно широк и его достаточно для описания большинства ограничений, которые могут быть сформулированы при проектировании.

### 7.3 Сервисы и потоки работ

Как было отмечено выше, подход СИНТЕЗа подразумевает возможность регистрации сервисов и потоков работ. Для решения такой задачи представленных методов и средств недостаточно. Для сервисов и потоков работ необходимо разработать специфические техники сопоставления и техники формирования взглядов.

## 8 Заключение

В данной работе рассмотрены проблемы и некоторые результаты исследования задачи отображения спецификаций как подзадачи, возникающей при регистрации информационных ресурсов в предметном посреднике. Приведен обзор существующих решений и выделены преимущества подхода СИНТЕЗа к отображению спецификаций, выраженных в канонической информационной модели. Основной вклад данной работы заключается в следующем:

- определена формальная постановка задачи отображения спецификаций в канонической информационной модели

- предложена модификация алгоритма формирования взглядов [4], расширяющая его на случай, когда один класс ресурса выражается при помощи композиции нескольких классов посредника
  - предложен метод автоматизации формирования условий во взглядах на основе анализа ограничений целостности, определенных в отображаемых спецификациях
  - реализован прототип системы поддержки процесса регистрации, реализующий описанный подход к отображению спецификаций ресурса и посредника с помощью техники GLAV
- Также, в работе приведены направления дальнейших исследований, с учетом которых будет развиваться описанный подход и прототип системы поддержки регистрации.

## Литература

- [1] Kalinichenko L.A., Stupnikov S.A., Martynov D.O. SYNTHESIS: a Language for Canonical Information Modeling and Mediator Definition for Problem Solving in Heterogeneous Information Resource Environments. Moscow: IPI RAN, 2007. - 171 p.
- [2] Kalinichenko L.A. Compositional Specification Calculus for Information Systems Development Advances in Databases and Information Systems: Proc. of the Third East European Conference. LNCS 1691. -- Berlin-Heidelberg: Springer-Verlag, 1999. -- P. 317--331.
- [3] Briukhov D.O., Kalinichenko L.A., Stupnikov S.A. Compositional approach for heterogeneous sources registration at a subject mediator. Emerging Database Research in Eastern Europe: Proc. of the Pre-Conference Workshop of VLDB 2003. -- Cottbus: Brandenburg University of Technology, 2003. -- P. 5-11.
- [4] Рябухин О.В., Брюхов Д.О., Калиниченко Л.А. Формирование выражений взглядов в задаче регистрации ресурсов в предметных посредниках. Труды 11-й Всероссийской Научной Конференции "Электронные библиотеки: Перспективные Методы и Технологии, Электронные коллекции" RCDL'2009. Петрозаводск: Петрозаводский государственный университет, 2009
- [5] Брюхов Д.О. Конструирование информационных систем на основе интероперабельных сред информационных ресурсов. Кандидатская диссертация. -- Москва: ИПИ РАН, 2003. -- 158 с.
- [6] Jeffrey D. Ullman, Information Integration Using Logical Views, Proceedings of the 6th International Conference on Database Theory, p.19-40, January 08-10, 1997
- [7] Miller R.J., Haas L., Hernandez M.A. Schema mapping as query discovery. In: Proc 26th International Conference On Very Large Data Bases (VLDB), 2000. p. 77--88

- [8] P. Shvaiko, J. Euzenat. A survey of schema-based matching approaches. *Journal on Data Semantics*, IV:146-171, 2005.
- [9] A. Deutsch, A. Nash, J. Remmel. The chase revisited, *Proceedings of the twenty-seventh ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 2008.
- [10] Скворцов Н.А. Связывание объектных спецификаций по семантике онтологического уровня. Труды 8-й Всероссийской Научной Конференции "Электронные библиотеки: Перспективные Методы и Технологии, Электронные коллекции" RCDL'2006. Ярославль: Ярославский государственный университет им. П.Г. Демидова, 2006. С. 70-74.
- [11] Synthesis Project (<http://synthesis.ipi.ac.ru/synthesis/projects/SYNTHESIS>)
- [12] Clio Project (<http://dblab.cs.toronto.edu/project/clio>)
- [13] AutoMed Project (<http://www.doc.ic.ac.uk/automed>)
- [14] Eclipse Project (<http://www.eclipse.org>)
- [15] Eclipse Graphical Editing Framework (<http://www.eclipse.org/gef>)
- [16] Eclipse Modeling Framework Project (<http://www.eclipse.org/emf>)

### **Semiautomatic GLAV mapping of specifications in a canonical information model of subject mediators**

Ryabukhin O. V.

Subject mediator is a component of the middleware located between resources and users (application) to provide a unified access to multiple resources registered in it. Subject mediator is constructed for solution of a class of problems in particular domain (in such scientific fields as, for example, the Earth sciences, astronomy, biology, and others). This work is devoted to semi-automatic construction of GLAV (Global-Local-As-View) mappings between specifications of information resources and mediator. We define the problem of specifications mapping as a part of the resource registration process and describe the idea of semiautomatic solution for this problem; we provide an overview of existing solutions and present a prototype of the registration process support system, which implements the described approach to semiautomatic specifications mapping.

---

Работа поддержана грантами РФФИ 08-07-00157-а, 10-07-00342-а, 10-07-00640-а и Программой фундаментальных исследований Президиума РАН № 3 *Фундаментальные проблемы системного программирования*, раздел 4 - Системы управления базами данных, проект *Исследование методов и средств промежуточного слоя предметных посредников, обеспечивающего решение*