

Формирование выражений взглядов в задаче регистрации ресурсов в предметных посредниках*

© Рябухин О.В.

Брюхов Д.О.

Калиниченко Л.А.

Институт проблем информатики РАН
{ovr, brd, leonidk}@ipi.ac.ru

Аннотация

Целью данной работы является развитие технологии регистрации информационных ресурсов в предметных посредниках. Основное внимание уделено заключительному этапу данного процесса – формированию выражений взглядов. Этот этап рассматривается как полуавтоматический процесс, подразумевающий участие эксперта в ходе принятия решений. В работе рассмотрена возможная его организация, позволяющая системе регистрации предложить эксперту возможные решения.

1 Введение

1.1 Концепция предметных посредников

В настоящее время в мире наблюдается экспоненциальный рост количества информации, накапливаемой в различных предметных областях в неоднородных распределенных информационных ресурсах (базах данных, сервисах, процессах). При решении задач в этих областях появляется необходимость совместного использования таких ресурсов, их интеграции. Концепция предметных посредников ориентирована на поддержку решения задач над множеством неоднородных информационных ресурсов. В качестве канонической информационной модели в рассматриваемой архитектуре предметных посредников используется язык СИНТЕЗ [6]. Занимая место между пользователем и ресурсами, предметный посредник предоставляет общий интерфейс доступа к ним; использование канонической информационной модели позволяет преодолеть модельную неоднородность ресурсов, а исчисление спецификаций дает возможность проведения интеграции спецификаций ресурсов в спецификациях посредника. Концепция интеграции ресурсов рассматривает два подхода, позволяющих

решать задачи над множеством неоднородных ресурсов: подход, движимый задачей (приложением), и подход, движимый ресурсами. В подходе, движимом ресурсами, глобальная схема строится как композиция множества схем имеющихся информационных ресурсов. Недостатком такого решения является плохая масштабируемость, т.к. при появлении очередного ресурса глобальная схема требует изменений. В основе концепции предметных посредников лежит движение от задачи к ресурсам. При подобном подходе, глобальная схема (концептуальная схема предметной области) строится независимо от конкретных информационных ресурсов и утверждается экспертным сообществом в данной предметной области. Также в архитектуре предметных посредников используются техники виртуальной интеграции, при которых виртуальные классы глобальной схемы представляют собой взгляды [7] над классами ресурсов (техника Global-as-View (GAV)), либо классы ресурсов являются взглядами над виртуальными классами глобальной схемы (техника Local-as-View (LAV)). Подходу предметных посредников, в котором глобальная схема не зависит от схем ресурсов, органично соответствует техника LAV, при которой классы ресурса выражаются как композиция классов посредника. Следует, однако, особо отметить, что при интеграции спецификаций неизбежно возникают различного рода конфликты. Конфликты могут возникать как из-за разных прикладных областей, так и из-за различного представления спецификаций семантически близких сущностей. Для устранения рассогласований используются методы структурных преобразований, а также метод использования функций разрешения конфликтов, описанных на языке высокого порядка. Очевидно, что такие функции должны соответствовать движению результатов запросов от ресурса к посреднику, и задавать соответствующие преобразования. Следовательно, функции разрешения конфликтов удобно разместить в теле GAV – правила. С учетом этого, использование техники GLAV (Global-Local-as-View) предпочтительно. GLAV соединяет в себе преимущества техник GAV и LAV, предоставляя возможность использования конструкций,

соответствующих классам ресурса, в голове LAV – правила, позволяя, таким образом, формировать выражения GAV.

1.2 Процесс регистрации

Выражения взглядов формируются в заключительной стадии так называемого процесса регистрации информационных ресурсов в предметном посреднике. Процесс регистрации [4] представляет собой последовательность согласованных действий эксперта и системы регистрации ресурсов, в ходе которой происходит поиск ресурсов-кандидатов для регистрации по нефункциональным требованиям (поиск по метаданным), выбор среди них ресурсов, онтологически релевантных посреднику, сопоставление спецификаций ресурса и посредника, и, наконец, формирование выражений GLAV взглядов. Процесс рассматривается как полуавтоматический: на каждом его шаге система регистрации ресурсов предлагает эксперту возможное решение (например, список пар онтологически релевантных типов), после чего эксперт подтверждает (либо корректирует) предложенное решение.

1.3 Формирование выражений взглядов

К моменту написания данной работы, заключительный этап процесса регистрации – шаг формирования выражений взглядов – подразумевал полную его реализацию экспертом (в частности, в работе [3] приведены примеры взглядов, определенных экспертом вручную, по соотношениям между классами ресурса и посредника). В данной работе процесс формирования выражений взглядов (аналогично другим этапам процесса регистрации) также рассматривается как полуавтоматический, описывается возможная его организация, позволяющая системе регистрации предложить эксперту возможные решения.

2 Общие понятия

Конечной целью процесса регистрации, помимо формирования выражений взглядов, является достижение отношения уточнения между спецификациями ресурса и посредника. Говоря неформально, можно утверждать, что спецификация А уточняет спецификацию В, если пользователь не замечает подмены при использовании А вместо В. Очевидно, что не каждая спецификация типа ресурса уточняет спецификацию соответствующего типа посредника, поэтому в процессе регистрации необходимо произвести поиск необходимых фрагментов. Специальное композиционное исчисление спецификаций [5], которое формализует процесс композиции и декомпозиции их фрагментов, позволяет решать задачи в таких

условиях. Важнейшими здесь являются понятие отношения уточнения, понятия редукта типа и наиболее общего редукта двух типов, а также операции композиции типов.

Определение. 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. Наиболее 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. Операция meet. Операция образует тип T как пересечение спецификаций типов-операндов. Тип T образуется слиянием двух наиболее 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 операций. Если в объединении имеется 2 операции 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 наиболее общих редуктов.

Определение. 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 наиболее 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 операций. Если в объединении имеется 2 операции 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 .

3 Процесс регистрации

В общих чертах, процесс регистрации сводится к следующим основным шагам:

1. Поиск информационных ресурсов по нефункциональным требованиям к ним (поиск по метаданным)
2. Поиск среди найденных кандидатов для регистрации ресурсов, онтологически релевантных посреднику

3. Идентификация типов посредника, онтологически релевантных типам ресурса.

4. Построение наиболее общих редуктов для каждой пары релевантных типов. Если при этом возникают конфликты между спецификациями, на этом шаге формируются функции разрешения конфликтов

5. Формирование взглядов

На этапах 1-3 осуществляется отбор ресурсо-кандидатов для регистрации в посреднике, при этом используются методы онтологического поиска, а также поиск по метаданным. Далее в работе подразумевается, что для регистрации выбран конкретный информационный ресурс, и, с помощью онтологических методов, получен набор попарно релевантных типов ресурса и посредника.

В работе [1] показаны возможные пути автоматизации шага 4 процесса регистрации. Приведен алгоритм построения наиболее общего редукта, основанный на проверке формального определения уточнения для спецификаций типов. Рассмотрены методы автоматизации разрешения конфликтов, основанные на применении правил структурных преобразований и использовании функций разрешения конфликтов, описываемых на языке высокого порядка. Техника применения правил структурных преобразований основывается на методе релевантных путей. Функции разрешения конфликтов описываются на языке Syfs (Synthesis Formula Subset), который является вариантом типизированного языка логики первого порядка.

4 Формирование выражений взглядов

Процесс формирования выражений взглядов представляет собой последовательность согласованных действий эксперта и системы регистрации ресурсов. Используемое далее в описании структуры процесса понятие дерева типа заимствовано из работы [1] и вводится следующим образом:

Определение. Дерево типа T строится по следующим правилам:

1. корнем дерева является тип T
2. вершинами дерева уровня $i+1$ являются типы $\{T_{i+1}\}$ непосредственно связанные с произвольным типом T_i уровня i посредством атрибута-ссылки $(T_i-a \rightarrow T_{i+1})$ и не содержащиеся в пути между типами T и T_i
3. ребрами дерева являются атрибуты-ссылки
4. листьями дерева являются атрибуты-значения типов

При построении дерева для устранения возможных циклов, если очередной тип, возникающий при построении дерева, уже встречался в пути к нему от вершины дерева, то новой вершины не вводится, и соответствующий атрибут-ссылка не рассматривается. Если же тип еще не встречался в пути, независимо от того, встречается он в дереве или нет, вводится новая вершина (являющаяся, таким образом,

дублирующей, если такой тип уже встречался в дереве). Т.е. каждый раз при появлении очередной вершины вводится новый набор ее вершин-потомков, что исключает образование циклов.

Дополнительных пояснений требует формирование наиболее общих редуктов, при котором некоторым элементам спецификации типа T_1 требуется ставить в соответствие специальные функции разрешения конфликтов, включаемые в спецификацию сопряженного редукта. Таким образом происходит преобразование сопряженных редуктов в конкретизирующие.

Далее структура процесса формирования взглядов рассматривается с точки зрения системы регистрации, действия эксперта и комментарии выделены курсивом (во всех случаях вмешательства эксперта без особых указаний предполагается, что его действия включают подтверждение или корректировку предложенного системой решения).

Для каждого класса ресурса:

1. Взять тип экземпляров данного класса (тип А)
2. Найти релевантные ему типы посредника, которые являются типами экземпляров каких-либо классов посредника

3. Выдать результат эксперту

Полученный экспертом набор релевантных типов используется для идентификации релевантных классов так, что если тип ресурса релевантен типу посредника, и данные типы являются типами экземпляров каких-либо классов, то эти классы возможно релевантны

4. Для каждой пары таких типов (тип А – релевантный ему тип посредника), формировать выражения взглядов, определяющие соответствующий типу ресурса класс ресурса через соответствующий типу посредника класс посредника:

5. Построить деревья типа ресурса (типа А) и релевантного ему типа посредника (типа В)

6. В построенных деревьях типов для каждой пары релевантных типов построить возможный вариант их наиболее общего редукта, сформировать возможные функции разрешения конфликтов, выдать результат эксперту

Правильность автоматического построения наиболее общего редукта, а также функций разрешения конфликтов в случае структурных несогласований целиком зависит от правильности установления отношения релевантности между типами ресурса и посредника, что требует тщательной проверки экспертом. Автоматическое построение функций разрешения конфликтов в случае сложных зависимостей не представляется возможным.

7. Для последующей типизации голов GAV и LAV правил [2], сформировать композиционный тип соединением построенных ранее редуктов (композиционный тип С) и соответствующих им конкретизирующих редуктов (композиционный тип

D). При построении композиционного типа использовать переименование атрибутов во избежание конфликтов имен. Выдать результат эксперту.

Полученный экспертом композиционный тип должен определять совокупность общих элементов спецификаций в деревьях типов ресурса и посредника, находящихся в отношении уточнения.

8. Сформировать голову правил GAV и LAV, типизированную полученным композиционным типом.

9. Сформировать тело GAV – правила:

10. Указать в нем соответствующий класс ресурса, типизированный композицией конкретизирующих редуктов (тип D), с добавлением соответствующих функций разрешения конфликтов

11. Сформировать тело LAV – правила:

12. Для каждого атрибута композиционного типа, не вошедшего в редукт типа посредника (типа B), произвести поиск пути к соответствующему атрибуту в его дереве. Выдать результат эксперту.

13. Если для класса ресурса получилось несколько GLAV – правил, произвести их композицию, выразив, в конечном итоге, класс ресурса через композицию классов посредника. Выдать результат эксперту.

Описанный процесс иллюстрируется на сокращенной адаптации примера, рассматриваемого в [2]. Предметной областью являются задачи поиска далеких галактик [2], которые были сформулированы в контексте подхода, развиваемого на основе виртуальных обсерваторий (Virtual Observatory, VO). Далее в примере рассматривается единственный информационный ресурс SDSS (Sloan Digital Sky Survey), содержащий обзор части видимого звездного неба.

Схема посредника включает в себя следующие типы (здесь и далее используется нотация языка СИНТЕЗ [6]):

```

{CoordEQJ; in: type;
  ra: real;
  de: real;
}
{CatalogData; in: type;
  name: string;
  coord: CoordEQJ;
}
{Magnitude; in: type;
  magValue: real;
  magError: real;
  filter: string;
}
{OpticalCatalogData; in: type; supertype:
CatalogData;
  colorIndexURG: real;
  deltaColorIndexURG: real;
  magnitude: {set; type_of_element:
Magnitude;};
};

```

Тип CatalogData описывает данные, накапливаемые в различных каталогах астрономических объектов, атрибут name определяет имя объекта, атрибут coord – координаты: атрибут ra типа CoordEQJ соответствует прямому восхождению, de – склонению. Подтип OpticalCatalogData типа CatalogData описывает данные, накапливаемые в оптических каталогах. Атрибуты colorIndexURG и deltaColorIndexURG имеют семантику показателей цвета. Тип Magnitude соответствует понятию звездной величины.

Посредник содержит единственный класс (instance_section определяет тип экземпляров данного класса):

```

class_specification:
  {opticalCatalogData; in: class;
instance_section: OpticalCatalogData;
};

```

SDSS содержит фотометрический обзор неба в фильтрах u, g, r, i, z, его схема описана следующим образом:

```

{PhotoPrimary; in: type;
  objID: integer;
  ra: real;
  dec: real;
  u: real;
  err_u: real;
  g: real;
  err_g: real;
  i: real;
  err_i: real;
  r: real;
  err_r: real;
  z: real;
  err_z: real;
};

```

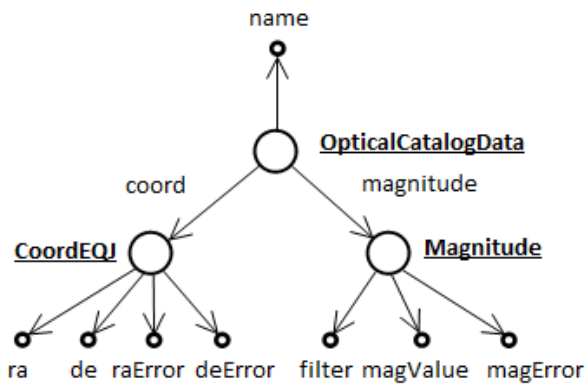
```

class_specification:
{photoPrimary; in: class;
  instance_section: PhotoPrimary;
};

```

Атрибут objID представляет собой идентификатор астрономического объекта, ra и dec определяют его координаты, остальные атрибуты соответствуют фотометрическим параметрам.

Опуская описание предыдущих шагов процесса регистрации, положим, что тип экземпляров класса photoPrimary ресурса PhotoPrimary, релевантен трем типам посредника: OpticalCatalogData, CoordEQJ, Magnitude. Из этих трех типов, тип OpticalCatalogData является типом экземпляров класса посредника opticalCatalogData. Следовательно, единственный GLAV – взгляд будет выражать класс посредника photoPrimary через класс ресурса opticalCatalogData. Дерево типа PhotoPrimary состоит из вершины, соответствующей данному типу, и вершин, соответствующих атрибутам. Дерево типа OpticalCatalogData выглядит следующим образом:



В данных деревьях типов имеем три пары релевантных типов: PhotoPrimary ~ OpticalCatalogData, PhotoPrimary ~ CoordEQJ, PhotoPrimary ~ Magnitude. Заметим, что построение деревьев типов и поиск в них пар релевантных типов позволяет локализовать поиск общей информации между классами ресурса и посредника.

Далее для каждой пары релевантных типов строятся наиболее общий и соответствующий конкретизирующий редукты. Заметим, что для этого применяются уже упоминавшиеся выше алгоритм построения наиболее общих редуктов, методы структурных преобразований, задание функций разрешения конфликтов.

Наиболее общий редукт типов OpticalCatalogData и PhotoPrimary выглядит следующим образом:

```

{R_OData_SDSS; in: reduct;
 metaslot of: OpticalCatalogData;
 taking: {name, coord, magnitude};
 c_reduct: CR_OData_SDSS;
 end;
};

```

Ключевые слова of, taking, c_reduct указывают, что R_OData_SDSS является редуктом типа OpticalCatalogData, в нем остаются атрибуты name, coord и magnitude, и CR_OData_SDSS является соответствующим конкретизирующий редуктом.

```

{CR_OData_SDSS; in: c_reduct;
 metaslot of: PhotoPrimary;
 taking: {objID};
 reduct: R_OData_SDSS;
 end;
simulating: {
 R_OData_SDSS.name ~ CR_OData_SDSS.objID,
 R_OData_SDSS.coord ~
 CR_OData_SDSS.get_coord,
 R_OData_SDSS.magnitude ~
 CR_OData_SDSS.get_magnitude;
 get_coord: {in: function; params: { -returns/
 {set; type_of_elements: CoordEQJ}}};
 {{ returns' = this/CR_Coord_SDSS }}
};
 get_magnitude: {in: function; params: { -
 returns/ {set; type_of_elements:

```

```

Magnitude}}};
 {{ returns' = this/CR_Magnitude_SDSS }}
};
};
};

```

Здесь секция simulating указывает на отношение уточнения между соответствующими элементами (например, атрибут objID ресурса соответствует атрибуту name посредника). Аналогично строятся остальные редукты: R_Magnitude_SDSS, R_Coord_SDSS, и соответствующие им конкретизирующие редукты CR_Magnitude_SDSS, CR_Coord_SDSS. Далее необходимо выделить наибольшую совокупность элементов спецификаций, находящихся в отношении уточнения в построенных ранее деревьях типов. Такой наибольший фрагмент может быть описан с помощью специального композиционного типа, полученного путем применения операции join к полученным ранее редуктам и соответствующим им конкретизирующим редуктам с соответствующими переименованиями.

```

CT_OData[name, coord, coord_ra, coord_de,
 magnitude, magnitude_magValue,
 magnitude_magError, magnitude_filter] =
 R_OData_SDSS[name, coord, magnitude] |
 R_Magnitude_SDSS[magnitude_magValue:
 magValue, magnitude_magError:magError,
 magnitude_filter:filter] |
 R_Coord_SDSS[coord_ra:ra, coord_de:de]

```

```

CT_SDSS[objID, get_coord, get_magnitude, u,
 err_u, g, err_g, i, err_i, r, err_r, z,
 err_z, get_magValue, get_magError,
 get_filter, ra, dec] = CR_OData_SDSS[objID,
 get_coord, get_magnitude] |
 CR_Magnitude_SDSS[u, err_u, g, err_g, i,
 err_i, r, err_r, z, err_z, get_magValue,
 get_magError, get_filter] |
 CR_Coord_SDSS[ra, dec]

```

Идентификатор типа с добавленными квадратными скобками специфицирует редукт данного типа, включающий атрибуты, перечисленные в скобках. Вертикальная черта обозначает операцию join композиции типов.

Выражения взглядов формулируются с помощью формул на языке Syfs, который является вариантом типизированного языка логики первого порядка. Предикаты в формулах соответствуют классам и функциям. Правила имеют вид:

$$Q(v/T_v):-C_1(v_1/T_{v_1}),\dots,C_n(v_n/T_{v_n}),F_1(t_1,y_1),\dots,F_m(t_m,y_m),B$$

где C_i – предикат класса, F_i – функциональный предикат, B – ограничение. Правила представляют собой конъюнкцию предикатов и функций. Переменные v, v_1, \dots, v_n , типизированы T_v, \dots, T_{v_n} соответственно.

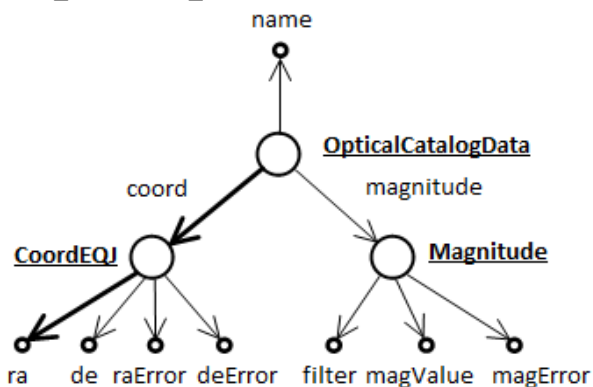
Определим голову взглядов GAV и LAV: она типизируется композиционным типом CT_OData:

```
v_SDSS_OData(x/CT_OData[name, coord,
coord_ra, coord_raError, coord_de,
coord_deError, magnitude,
magnitude_magValue,
magnitude_magError, magnitude_filter])
```

Для формирования тела GAV – взгляда допишем в него класс ресурса, типизированный редуктом композиционного типа CT_SDSS, включающим в себя все необходимые атрибуты, а также перечислим функции разрешения конфликтов, полученные при построении конкретизирующих редуктов.

```
v_SDSS_OData(x/CT_OData[name, coord,
coord_ra, coord_raError, coord_de,
coord_deError, magnitude,
magnitude_magValue,
magnitude_magError, magnitude_filter])
:- photoPrimary(x/CT_SDSS[name:objID,
coord_ra: ra, coord_de: dec])
& get_coord(x, coord)
& get_magnitude(x, magnitude)
& get_magValue(x, magnitude_magValue)
& get_magError(x, magnitude_magError)
& get_filter(x, magnitude_filter)
```

Для построения тела LAV – правила, допишем в него класс посредника, типизированный редуктом типа его экземпляров (в данном случае класс opticalCatalogData, тип его экземпляров OpticalCatalogData). Для каждого атрибута композиционного типа, не вошедшего в тип OpticalCatalogData, ищем путь к соответствующему атрибуту в дереве типа. Например, для атрибута coord_ra типа CT_OData:



Следовательно, в редукте типа OpticalCatalogData атрибут, соответствующий coord_ra, может быть найден по пути coord.ra. В итоге:

```
v_SDSS_OData(x/CT_OData[name, coord,
coord_ra, coord_de, magnitude,
magnitude_magValue, magnitude_magError,
```

```
magnitude_filter]):-
opticalCatalogData(x/OpticalCatalogData
[name, coord, magnitude,
coord_ra:coord.ra, coord_de:coord.de,
magnitude_magValue: magnitude.magValue,
magnitude_magError: magnitude.magError,
magnitude_filter: magnitude.filter])
```

5 Заключение

Концепция предметных посредников позволяет создать инструментарий для решения задач над множеством неоднородных информационных ресурсов. В данной работе были затронуты вопросы, связанные с технологией регистрации ресурсов в посредниках. Заключительному шагу данного процесса уделено особое внимание. Формирование выражений взглядов определено как полуавтоматический процесс, представляющий собой последовательность согласованных действий эксперта и системы регистрации ресурсов. В результате выполнения определенных шагов система предоставляет эксперту возможное решение, которое может быть подтверждено, или скорректировано. Для построения возможных решений системой рассмотрен подход, в основе которого лежит использование соответствий между спецификациями ресурсов и посредника, полученных в результате построения конкретизирующих редуктов. Для локализации поиска релевантных типов в рамках релевантных классов и поиска путей при формировании LAV – правил используются деревья типов. Предлагаемый подход позволяет включить процесс формирования выражений взглядов в общий процесс регистрации для согласования действий эксперта и системы регистрации ресурсов.

Литература

- [1] Брюхов Д.О. Конструирование информационных систем на основе интероперабельных сред информационных ресурсов. -- Москва: ИПИ РАН, 2003. -- 158 с.
- [2] 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.
- [3] 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. -- Pereslavl-Zalesskij: Pereslavl University, 2007. -- P. 253--262.

- [4] 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.
- [5] 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.
- [6] 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.
- [7] Jeffrey D. Ullman, Information Integration Using Logical Views, Proceedings of the 6th International Conference on Database Theory, p.19-40, January 08-10, 1997

Views expressions construction at information resource registration in typed subject mediator

Ryabukhin O. V., Briukhov D. O., Kalinichenko L.A.

This paper is focused on evolution of information resources registration process in a typed subject mediator. Final phase of this process - the process of views expressions construction is specially emphasized. This phase is considered to be a semi-automatic process with expert participation required. Possible organization of this process, which allows the resource registration system to suggest possible solutions, is described.

* Работа выполнена при финансовой поддержке РФФИ (проект 08-07-00157-а) и Программы фундаментальных исследований Президиума РАН № 3 "Фундаментальные проблемы системного программирования" (проект "Исследование методов и средств промежуточного слоя предметных посредников, обеспечивающего решение задач над множеством неоднородных распределенных информационных ресурсов").