

УДК 004.6

ОБОБЩЕННАЯ ОПЕРАЦИЯ АБСТРАГИРОВАНИЯ КАК РЕАЛИЗАЦИЯ ПРИНЦИПА ОТКРЫТОСТИ САМООРГАНИЗУЮЩЕЙСЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ

*В.В. Дрождин, к.т.н.; Р.Е. Зинченко; Е.В. Герасимова
(Пензенский государственный педагогический университет им. В.Г. Белинского,
drozhdin@yandex.ru)*

Рассмотрен принцип открытости самоорганизующейся информационной системы на уровне концептуального моделирования предметной области. Предложена реализация открытости на основе обобщенной операции абстрагирования, позволяющей формировать образы объектов, обладающие большой выразительной способностью. На примерах показано формирование образов с отбрасыванием несущественных свойств, интегральными и усредненными свойствами и др.

Ключевые слова: информационная система, самоорганизующаяся информационная система, моделирование предметной области, концептуальная модель предметной области, отношение абстрагирования.

Компьютеры, обладая огромными ресурсами, стремительно проникают в различные сферы деятельности человека. Это требует формализации специфических предметных областей (ПрО) и разработки методов решения все более сложных задач. Однако невозможно быстро и сразу создавать хорошие информационные системы для отражения сложных ПрО, поэтому они проектируются и создаются поэтапно, что часто требует доработки уже используемых подсистем. Более того, в [1] отмечено, что всякую ПрО можно в любой момент уточнить новыми подробностями, отличительными признаками, вариантами и прочими дополнениями, и в этом смысле она бесконечна. Следовательно, все попытки окончательно формализовать какую бы то ни было ПрО обречены на провал. С другой стороны, в процессе эксплуатации автоматизированной информационной системы (АИС) постепенно уменьшается ее адекватность внешней среде вследствие изменения среды (состава и качества пользователей) и решаемых задач, однако при этом модель ПрО в информационной системе остается фиксированной. Это снижает качество удовлетворения информационных потребностей пользователей и требует модификации системы. Следовательно, необходимы новые принципы построения и подходы к созданию дружественных (ориентированных на конечных пользователей) АИС. На взгляд авторов, наиболее продуктивный подход к организации систем нового типа определила синергетика, исследующая самоорганизующиеся системы, основными характеристиками которых являются эволюционные изменения, открытость, нелинейность функционирования и наличие хаоса (флуктуаций), являющегося фундаментальным регулятором устойчивости и развития систем. В области программного обеспечения эти принципы выражены в [1]: в качестве новых аспектов взаимодействия выступают время и открытость. В силу своей открытости система может получать от окружения неалгоритмические стимулирующие воздействия, и, соответственно,

ее поведение также принимает неалгоритмический характер.

Самоорганизующаяся информационная система (СИС) является открытой на всех уровнях своей организации: открытость структур, открытость взаимодействия с внешней средой и т.д. Для поддержки адекватности внешней среде СИС должна адаптироваться к ее изменениям путем все более полного учета семантики ПрО. Разработчики АИС не в состоянии предвидеть и реализовать в системе все особенности той или иной ПрО, поэтому СИС должна предоставить пользователям высокоуровневый инструментарий, позволяющий легко совершенствовать и расширять поддерживаемую в системе модель ПрО. В настоящей работе предлагается реализовать данный механизм на основе обобщенной операции абстрагирования.

Абстрагирование устанавливает отношение между понятием-образом и конкретным понятием [2]. Оно позволяет формировать объекты понятия-образа путем огрубления объектов конкретного понятия. Учитывая, что отношения агрегации, классификации и обобщения [3] также являются определенными видами абстрагирования, будем считать, что в рассматриваемом случае отношение абстрагирования реализуется способами формирования образов, отличными от способов реализации указанных отношений. Например, формирование понятия-образа может осуществляться:

- отбрасыванием несущественных свойств конкретного понятия;
- отбрасыванием свойств конкретного понятия, существенных для конкретных объектов, но не существенных в определенной ситуации, соответствующей понятию-образу;
- созданием обобщенного понятия с интегральными характеристиками, формируемыми на основе свойств конкретного понятия;
- созданием обобщенного понятия-образа, объекты которого будут иметь усредненные характеристики групп объектов конкретного понятия и др.

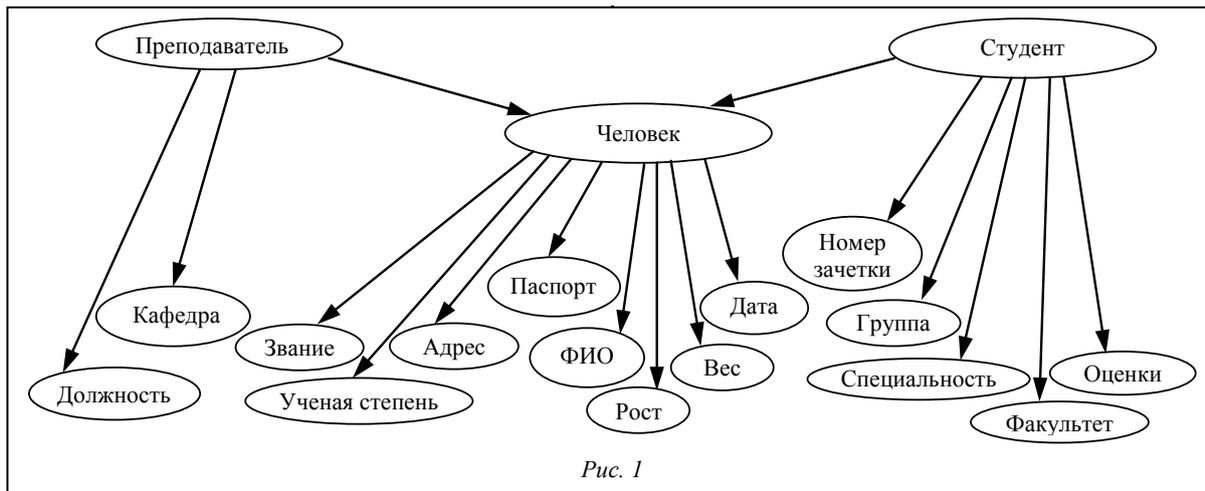


Рис. 1

Таким образом, отношение абстрагирования устанавливается между понятием-образом и конкретным понятием путем задания пользователем правила формирования объектов понятия-образа на основе объектов и свойств конкретного понятия.

Для демонстрации возможностей операции абстрагирования приведем примеры модификации концептуальной модели ПрО понятиями, создаваемыми с помощью этой операции. Для формализации отображения концептуальной модели ПрО в схему БД будем использовать язык описания модели ПрО и механизм соответствия, рассмотренные в [3, 4].

В качестве наглядной иллюстрации возьмем фрагмент ПрО «Учебный процесс вуза», основу которого составляет система понятий, представленная на рисунке 1, где стрелками обозначено отношение агрегации.

На рисунке 2 приведена схема БД для модели ПрО.

Для каждого понятия модели ПрО создается базовый SQL-запрос, позволяющий получить из БД множество объектов, соответствующих понятию, вида

```
CREATE OR REPLACE VIEW v_V AS
Select *
From V
```

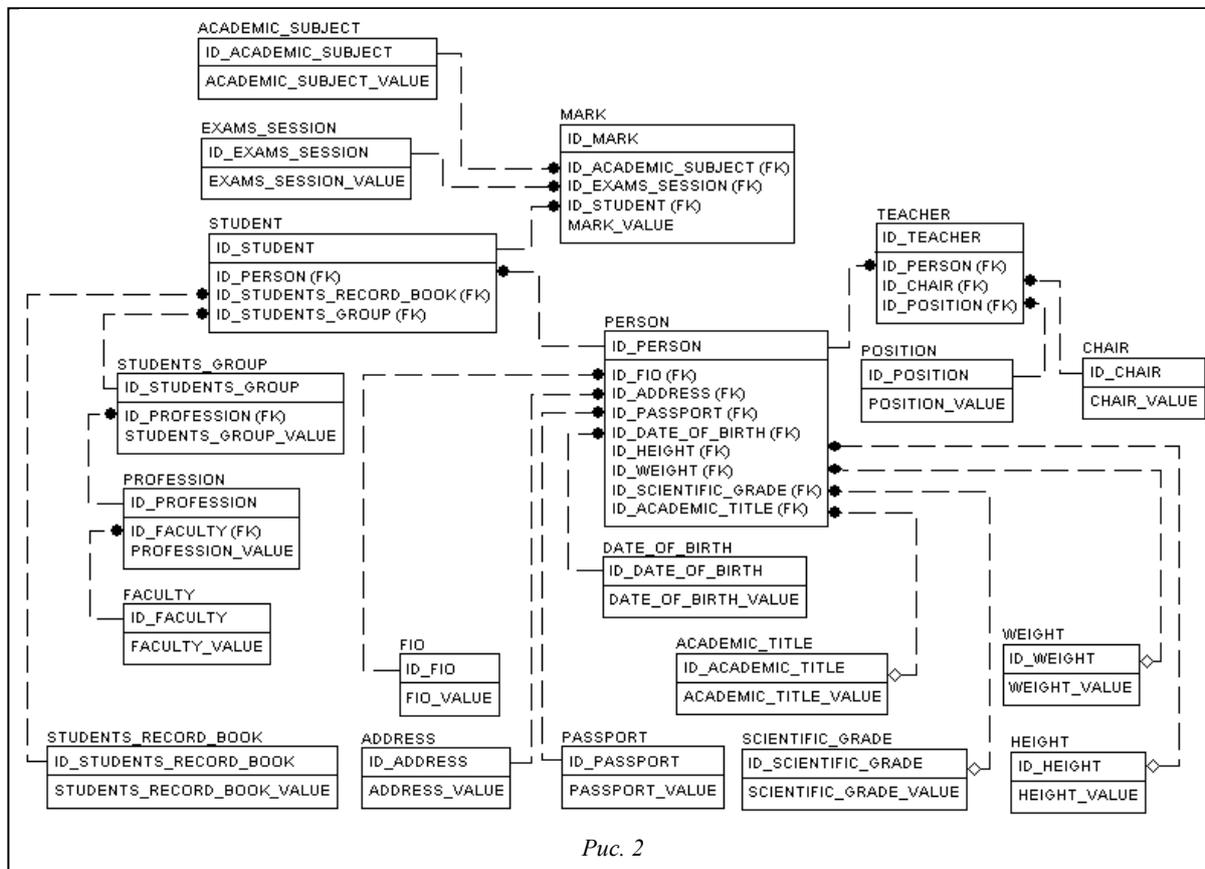


Рис. 2

где **V** – имя понятия модели ПрО в терминах схемы БД; **v_V** – наименование базового SQL-запроса для понятия **V**.

Например, базовый SQL-запрос для понятия «Факультет» (**FACULTY**) имеет вид
CREATE OR REPLACE VIEW v_FACULTY AS
Select *
From FACULTY

Для всех остальных понятий модели ПрО базовые SQL-запросы создаются аналогично.

Рассмотрим модификацию модели ПрО путем добавления новых понятий с организацией связи на основе отношения абстрагирования.

При этом возможны следующие варианты:

1) добавление нового понятия-образа, связываемого с существующим конкретным понятием модели ПрО;

2) добавление нового конкретного понятия, связываемого с существующим понятием-образом модели ПрО.

Например, в концептуальной модели ПрО (рис. 1) понятие «человек» содержит большое количество свойств, которые являются несущественными или отсутствуют (например, звание и ученая степень у студентов) у объектов понятий «студент» и «преподаватель». Поэтому информацию о человеке в понятиях «студент» и «преподаватель» целесообразно корректно представить с помощью промежуточных понятий-образов «человек-студент» и «человек-преподаватель», полученных абстрагированием понятия «человек» путем отбрасывания несущественных свойств. Добавление понятий-образов такого вида показано на рисунке 3, где для обозначения отношения абстрагирования используем запись $V_1 \bullet \dashrightarrow V_2$, означающую, что понятие V_1 является понятием-образом, а понятие V_2 – конкретным понятием для V_1 .

Включение в концептуальную модель ПрО понятия-образа такого типа не требует изменения схемы БД.

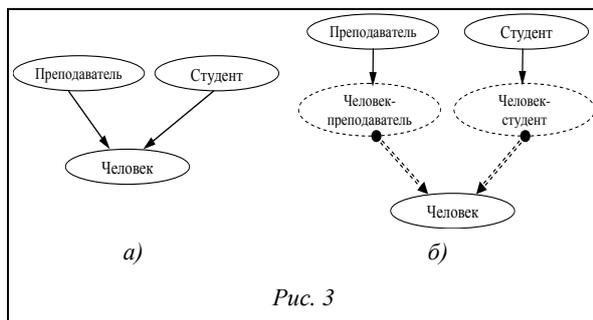


Рис. 3

Исходный фрагмент модели ПрО, показанный на рисунке 3а, формально представляется записью
 $V_1 = \langle \text{человек}, \langle : \text{ФИО}, : \text{паспорт}, : \text{дата рождения}, : \text{адрес}, : \text{ученая степень}, : \text{звание}, : \text{рост}, : \text{вес} \rangle, \emptyset \rangle$
 $V_2 = \langle \text{студент}, \langle : \text{человек}, : \text{зачетная книжка}, : \text{группа}, : \text{специальность}, : \text{факультет} \rangle, \emptyset \rangle$
 $V_3 = \langle \text{преподаватель}, \langle : \text{человек}, : \text{кафедра}, : \text{должность} \rangle, \emptyset \rangle$

$r_1 = \langle V_2, V_1, \text{является студентом, агрегация} \rangle$

$r_2 = \langle V_3, V_1, \text{является преподавателем, агрегация} \rangle$

После включения понятий-образов «человек-студент» и «человек-преподаватель» фрагмент модели ПрО на рисунке 3б формально будет представлен записью

$V_1 = \langle \text{человек}, \langle : \text{ФИО}, : \text{паспорт}, : \text{дата рождения}, : \text{адрес}, : \text{ученая степень}, : \text{звание}, : \text{рост}, : \text{вес} \rangle, \emptyset \rangle$

$V_2 = \langle \text{студент}, \langle : \text{человек-студент}, : \text{зачетная книжка}, : \text{группа}, : \text{специальность}, : \text{факультет} \rangle, \emptyset \rangle$

$V_3 = \langle \text{преподаватель}, \langle : \text{человек-преподаватель}, : \text{кафедра}, : \text{должность} \rangle, \emptyset \rangle$

$V_4 = \langle \text{человек-студент}, \langle : \text{человек} \langle : \text{ФИО}, : \text{паспорт}, : \text{дата рождения}, : \text{адрес} \rangle \rangle, \emptyset \rangle$

$V_5 = \langle \text{человек-преподаватель}, \langle : \text{человек} \langle : \text{ФИО}, : \text{паспорт}, : \text{дата рождения}, : \text{адрес}, : \text{ученая степень}, : \text{звание} \rangle \rangle, \emptyset \rangle$

$r_1 = \langle V_4, V_1, \text{является человеком-студентом, абстрагирование} \rangle$

$r_2 = \langle V_5, V_1, \text{является человеком-преподавателем, абстрагирование} \rangle$

$r_3 = \langle V_2, V_4, \text{является студентом, агрегация} \rangle$

$r_4 = \langle V_3, V_5, \text{является преподавателем, агрегация} \rangle$

Изменение системы базовых SQL-запросов будет заключаться в создании SQL-запросов для понятий-образов и модификации SQL-запросов для понятий, имеющих какую-либо связь с понятиями-образами.

Формирование базового SQL-запроса для понятия-образа производится путем выполнения операции проекции базового SQL-запроса, формирующего объем конкретного понятия (SQL-запросы, приведенные в статье, соответствуют синтаксису СУБД Oracle 10g [5, 6]):

1) человек-студент:

CREATE OR REPLACE VIEW v_PERSON_STUDENT AS
Select FIO_VALUE, PASSPORT_VALUE, DATE_OF_BIRTH_VALUE, ADDRESS_VALUE
From v_PERSON

2) человек-преподаватель:

CREATE OR REPLACE VIEW v_PERSON_TEACHER AS
Select FIO_VALUE, PASSPORT_VALUE, DATE_OF_BIRTH_VALUE, ADDRESS_VALUE, SCIENTIFIC_GRADE_VALUE, ACADEMIC_TITLE_VALUE
From v_PERSON

В общем случае базовый SQL-запрос для понятия-образа имеет вид

Select X1, X2, ..., Xn
From v_V

где **X1, X2, ..., Xn** – свойства, которыми обладают объекты понятия-образа; **v_V** – базовый SQL-запрос, формирующий объем конкретного понятия.

Модификация SQL-запросов для понятий, использующих образы, заключается в естественном соединении отношения, содержащего информацию об объектах этого понятия, и базового SQL-запроса, формирующего объем понятия-образа. Например, базовый SQL-запрос для понятия «студент» представляется в следующем виде:

CREATE OR REPLACE VIEW v_STUDENT AS
Select *
From STUDENT Natural Join v_PERSON_STUDENT

Аналогично базовый SQL-запрос для понятия «преподаватель» представляется в виде:

```
CREATE OR REPLACE VIEW v_TEACHER AS
```

```
Select *
```

```
From TEACHER Natural Join v_PERSON_TEACHER
```

С другой стороны, огрубление свойств может осуществляться усреднением их значений для множества однотипных объектов. При этом следует отметить, что возможны следующие варианты усреднения: логическое, арифметическое, статистическое и др.

Например, в концептуальной модели ПрО, содержащей понятие «преподаватель», создадим понятия-образы «кандидат на должность доцента» и «кандидат на должность профессора» путем логического усреднения значений определенных свойств конкретного понятия. Тогда понятие «кандидат на должность доцента» будет включать всех преподавателей, не работающих в должности доцента:

- докторов, кандидатов наук и не имеющих степени,

- имеющих звание доцента и не имеющих звания,

- имеющих необходимый преподавательский стаж или не имеющих его и др.,

а понятие «кандидат на должность профессора» будет включать всех преподавателей, не работающих в должности профессора: докторов и кандидатов наук, а также имеющих звание профессора или доцента и др.

Включение в концептуальную модель ПрО понятия-образа данного типа также не требует изменения схемы БД.

Измененный фрагмент модели ПрО формально будет представлен записью

$$V_1 = \langle \text{преподаватель}, \langle : \text{ФИО}, : \text{паспорт}, : \text{дата рождения}, : \text{адрес}, : \text{ученая степень}, : \text{звание}, : \text{кафедра}, : \text{должность} \rangle, \emptyset \rangle$$

$$V_2 = \langle \text{кандидат на должность доцента}, \langle : \text{преподаватель} \rangle, \text{должность} \neq \langle \text{доцент} \rangle \text{ и } (\text{степень} = \langle \text{кандидат наук} \rangle \text{ или } \langle \text{доктор наук} \rangle) \text{ и } (\text{звание} = \langle \text{доцент} \rangle \text{ или } (\text{звание} = \emptyset \text{ и } \text{стаж} > 5 \text{ лет})) \rangle$$

$$V_3 = \langle \text{кандидат на должность профессора}, \langle : \text{преподаватель} \rangle, \text{должность} \neq \langle \text{профессор} \rangle \text{ и } (\text{степень} = \langle \text{кандидат наук} \rangle \text{ или } \langle \text{доктор наук} \rangle) \text{ и } (\text{звание} = \langle \text{доцент} \rangle \text{ или } \langle \text{профессор} \rangle) \rangle$$

$$r_1 = \langle V_2, V_1, \text{является кандидатом на должность доцента, абстрагирование} \rangle$$

$$r_2 = \langle V_3, V_1, \text{является кандидатом на должность профессора, абстрагирование} \rangle$$

Изменение системы базовых SQL-запросов будет заключаться в создании базовых SQL-запросов для понятий-образов путем добавления к базовому SQL-запросу конкретного понятия соответствующего предиката. Например,

1) кандидат на должность доцента:

```
CREATE OR REPLACE VIEW v_CANDIDATE_TO_ASSOCIATE_PROF AS
```

```
Select *
```

```
From v_TEACHER
```

```
Were POSITION_VALUE != 'Доцент' and SCIENTIFIC_GRADE_VALUE IN ('Кандидат наук', 'Доктор наук') and (ACADEMIC_TITLE_VALUE = 'Доцент' or (ACADEMIC_TITLE_VALUE=NULL and EXPERIENCE_VALUE> 5))
```

(атрибут **EXPERIENCE_VALUE** (стаж) отсутствует в схеме БД, поэтому предполагается, что он вычисляется на основании даты приема преподавателя на работу);

2) кандидат на должность профессора:

```
CREATE OR REPLACE VIEW v_CANDIDATE_TO_PROFESSOR AS
```

```
Select *
```

```
From v_TEACHER
```

```
Were POSITION_VALUE != 'Профессор' and SCIENTIFIC_GRADE_VALUE IN ('Кандидат наук', 'Доктор наук') and ACADEMIC_TITLE_VALUE IN ('Доцент', 'Профессор')
```

В общем случае базовый SQL-запрос для понятия-образа данного типа имеет вид

```
Select *
```

```
From v_V
```

```
Were P
```

где **P** – соответствующий предикат.

Еще один вариант абстрагирования заключается в формировании обобщенного образа со статистическими характеристиками определенного множества объектов конкретного понятия (статистическое усреднение). Например, сформируем образ со статистической оценкой успеваемости определенного контингента студентов.

Для формирования успеваемости студентов необходимо определить контингент студентов и полученные оценки за требуемые сессии, разбить его на категории в соответствии с полученными оценками и задать правила вычисления статистических оценок успеваемости.

Формирование объема понятия «успеваемость студентов» определяется предикатом **P**, задающим характеристики студентов, включаемых в выборку (например, выделение студентов некоторой группы, специальности, курса и др.), и требуемые сессии.

Разбиение контингента студентов на категории осуществляется в соответствии с имеющимися у студентов оценками, совпадающими с категорией успеваемости (например, имеющие тройки, обучающиеся только на четыре, обучающиеся на четыре и пять и др.).

Вычисление статистической оценки успеваемости по соответствующей категории задается формулой: количество студентов, соответствующих категории, деленное на общее количество студентов и умноженное на 100.

На основе этих данных система сформирует следующий базовый SQL-запрос для понятия-образа «успеваемость студентов»:

```
CREATE OR REPLACE VIEW v_STUDENTS_SUMMARY_RESULTS AS
```

```
WITH contingent AS (select * from v_MARK where P),
```

```
student_3 AS (select ID_STUDENT, MIN(MARK_VALUE) min_mark, MAX(MARK_VALUE) max_mark from contingent group by ID_STUDENT having min_mark = 3 and max_mark = 3),
```

```
student_3_4 AS (select ID_STUDENT, MIN(MARK_VALUE) min_mark, MAX(MARK_VALUE) max_mark from contingent group by ID_STUDENT having min_mark = 3 and max_mark = 4),
```

```
student_4 AS (select ID_STUDENT, MIN(MARK_VALUE) min_mark, MAX(MARK_VALUE) max_mark
```

```

from contingent group by ID_STUDENT having
min_mark = 4 and max_mark = 4),
student_4_5 AS (select ID_STUDENT, MIN(MARK_VA-
LUE) min_mark, MAX(MARK_VALUE) max_mark
from contingent group by ID_STUDENT having
min_mark = 4 and max_mark = 5),
student_5 AS (select ID_STUDENT, MIN(MARK_VA-
LUE) min_mark, MAX(MARK_VALUE) max_mark
from contingent group by ID_STUDENT having
min_mark = 5 and max_mark = 5),
count_all AS (select count(distinct ID_STUDENT) a from
contingent)
select
(select a from count_all) as cnt_all,
(select count(*) from student_3) as cnt_3,
(select count(*) from student_3) / (select a from count_
all) *100 as prc_3,
(select count(*) from student_3_4) as cnt_3_4,
(select count(*) from student_3_4) / (select a from count_
all) *100 as prc_3_4,
(select count(*) from student_4) as cnt_4,
(select count(*) from student_4) / (select a from count_
all) *100 as prc_4,
(select count(*) from student_4_5) as cnt_4_5,
(select count(*) from student_4_5) / (select a from count_
all) *100 as prc_4_5,
(select count(*) from student_5) as cnt_5,
(select count(*) from student_5) / (select a from count_
all) *100 as prc_5
FROM contingent where rownum = 1;

```

Дадим некоторые пояснения к приведенному запросу. Подзапрос **contingent** формирует контингент студентов в соответствии с условием **P** (например, если **P** заменить на выражение **FACULTY_VALUE = 'ФЭМИ' and ACADEMIC_SUBJECT_VALUE = 'Информатика' and EXAMS_SESSION_VALUE=1**, то контингент студентов будет включать только тех студентов факультета ФЭМИ, у которых в первой сессии был экзамен по информатике).

Следующие подзапросы формируют списки студентов из выделенного контингента, которые учатся:

- только на «удовлетворительно» – **student_3**,
- на «удовлетворительно» и «хорошо» – **student_3_4**,
- только на «хорошо» – **student_4**,
- на «хорошо» и «отлично» – **student_4_5**,
- только на «отлично» – **student_5**.

И наконец, результирующий запрос будет содержать 11 показателей (**cnt_all**, **cnt_3**, **prc_3**, **cnt_3_4**, **prc_3_4**, **cnt_4**, **prc_4**, **cnt_4_5**, **prc_4_5**, **cnt_5**, **prc_5**), отражающих количественную и процентную разбивку выделенного контингента студентов (**contingent**) по категориям успеваемости.

Разработанный метод позволяет на основе понятий-образов, представляющих собой агрегированные количественные оценки, создавать понятия-образы, выражающие обобщенные качественные характеристики большого количества групп однотипных объектов. Например, на основе полученных выше статистических оценок успеваемости заданного контингента студентов определим качество этого контингента с точки зрения обуче-

ния. Для этого сформируем понятие «качество контингента студентов».

Предположим, пользователь задает некоторое правило, на основании которого определяется содержание понятия-образа «качество контингента студентов», например: не менее 75 % студентов учатся только на оценку «5», а остальные на «4» или «5» – отличное качество контингента; не менее 85 % студентов учатся на оценки «4» или «5», а остальные имеют оценки «3» – хорошее качество контингента; не менее 95 % студентов учатся на оценки «3», «4» или «5» – удовлетворительное качество контингента; в противном случае – неудовлетворительное. Задаваемые таким образом правила могут быть произвольными и достаточно сложными. На основе этих данных система сформирует следующий базовый *SQL*-запрос для понятия-образа «качество контингента студентов»:

```

CREATE OR REPLACE VIEW v_QUALITY_STUD-
CONT AS
Select case
when prc_5 >= 75 and cnt_all - cnt_5 = cnt_4_5 +
cnt_4 then 'отличное качество'
when prc_5 + prc_4_5 + prc_4 >= 85 and cnt_all -
(cnt_5 + cnt_4_5 + cnt_4) = cnt_3 + cnt_3_4 then
'хорошее качество'
when prc_5 + prc_4_5 + prc_4 + prc_3_4 + prc_3 >= 95
then 'удовлетворительное качество'
else 'неудовлетворительное качество'
end as 'качество контингента студентов'
From v_STUDENTS_SUMMARY_RESULTS

```

Подобным образом может быть сформирован очень широкий набор понятий-образов, а также создана иерархия или сеть понятий-образов, задающая мощную обработку данных и предоставляющая пользователям данные с высокой степенью абстрагирования и обобщения.

Еще один вариант абстрагирования заключается в формировании понятия-образа с интегральными характеристиками определенного множества объектов конкретного понятия. Например, сформируем понятие-образ «вид диплома», создаваемый на основании интегральных показателей успеваемости студента за весь период обучения в вузе, включая оценки за государственные экзамены и за дипломный проект.

Пользователь задает правило, на основании которого определяется вид диплома для каждого студента:

- 1) красный диплом: процент оценок «отлично» за весь период обучения больше или равен 75 и количество оценок «удовлетворительно» за весь период обучения равен 0 и процент оценок «отлично» за государственные экзамены равен 100 и оценка за дипломный проект равна 5;
- 2) синий диплом – иначе.

На основе этих данных система сформирует следующий базовый *SQL*-запрос для понятия-образа «вид диплома»:

```

CREATE OR REPLACE VIEW v_DIPLOMA_TYPE AS
Select criteria.*,
case

```

```

when prc_5_study >= 75 and cnt_3_study = 0 and diploma_mark = 5 and prc_5_state_exams = 100 then
    'красный диплом'
else 'синий диплом'
end as 'вид диплома'
From (
    WITH stud AS (Select * From v_STUDENT s where
exists (select * from v_MARK where ID_STUDENT = s.ID_STUDENT and EXAMS_SESSION_VALUE = 'Государственный экзамен' and MARK_VALUE is not NULL) and exists (select * from v_MARK where ID_STUDENT = s.ID_STUDENT and ACADEMIC_SUBJECT_VALUE = 'Дипломный проект' and MARK_VALUE is not NULL))
Select FIO_VALUE,
((select count(*) from v_MARK where ID_STUDENT = stud.ID_STUDENT and MARK_VALUE = 5 and ACADEMIC_SUBJECT_VALUE != 'Дипломный проект' and EXAMS_SESSION_VALUE != 'Государственный экзамен') / (select count(*) from v_MARK where ID_STUDENT = stud.ID_STUDENT and ACADEMIC_SUBJECT_VALUE != 'Дипломный проект' and EXAMS_SESSION_VALUE != 'Государственный экзамен') * 100) as prc_5_study,
(select count(*) from v_MARK where ID_STUDENT = stud.ID_STUDENT and MARK_VALUE = 3 and ACADEMIC_SUBJECT_VALUE != 'Дипломный проект' and EXAMS_SESSION_VALUE != 'Государственный экзамен') as cnt_3_study,
(select MARK_VALUE from v_MARK where ID_STUDENT = stud.ID_STUDENT and ACADEMIC_SUBJECT_VALUE = 'Дипломный проект') as diploma_mark,
((select count(*) from v_MARK where ID_STUDENT = stud.ID_STUDENT and MARK_VALUE = 5 and EXAMS_SESSION_VALUE = 'Государственный экзамен') / (select count(*) from v_MARK where ID_STUDENT = stud.ID_STUDENT and EXAMS_SESSION_VALUE = 'Государственный экзамен') * 100) as prc_5_state_exams
From stud) criteria;

```

Пояснения к запросу. Подзапрос **stud** формирует список студентов, которые защитили диплом и сдали все государственные экзамены. Следующие подзапросы формируют критерии, на основании которых будет вычислен тип диплома для каждого студента:

- 1) процент оценок «отлично» за весь период обучения в вузе – **prc_5_study**,
- 2) количество оценок «удовлетворительно» за весь период обучения в вузе – **cnt_3_study**,
- 3) оценка за дипломный проект – **diploma_mark**,
- 4) процент оценок «отлично» за государственные экзамены – **prc_5_state_exams**.

Результирующий запрос будет содержать Ф.И.О. студента, вышеперечисленные критерии для каждого студента и вид полученного студентом диплома.

В общем случае для формирования образов может использоваться базовый *SQL*-запрос, представляющий собой систему взаимосвязанных *SQL*-запросов, включающую

1) основной запрос:

```

Select f1 as X1, f2 as X2, ..., fn as Xn
From v_V

```

Where P

2) дополнительные вычисления:

- a) Name1 Select f₁' as X'₁, f₂' as X'₂, ..., f_n' as X'_n
From #
Where P₁
- б) Name2 Select f₁" as X"₁, f₂" as X"₂, ..., f_n" as X"_n
From #
Where P₂
- ...

где **f_i** – некая функция формирования значений свойства **X_i**, которая может иметь следующий вид:

- Name** – имя свойства из понятия **v_V**;
- φ(v_V.X)** – аналитическая зависимость вычисления значений свойства **X_i** на основе значений свойств **X** понятия **v_V**;
- case** – значение свойства **X_i**, соответствующее выполнению определенного условия;
- #** – источник данных, сформированный предикатом **P** в основном запросе.

Рассмотрим примеры добавления в модель ПрО новых конкретных понятий для имеющегося понятия-образа.

Включение в концептуальную модель ПрО нового конкретного понятия с несущественными свойствами относительно понятия-образа предполагает включение понятия, содержащего как важные свойства понятия-образа, так и несущественные. Предположим, что в модели ПрО существует понятие «студент-отличник». Тогда определим конкретное понятие «студент» для понятия-образа «студент-отличник». При этом несущественными свойствами у понятия «студент» будут «оценки по дисциплинам» и «стипендия».

Измененный фрагмент ПрО формально представляется записью:

$$\begin{aligned}
 V_1 &= \langle \text{студент-отличник}, \langle \text{:человек}, \text{:зачетка}, \text{:группа}, \text{:специальность}, \text{:факультет} \rangle, \emptyset \rangle \\
 V_2 &= \langle \text{студент}, \langle \text{:человек}, \text{:зачетка}, \text{:группа}, \text{:специальность}, \text{:факультет}, \text{:оценки}, \text{:стипендия} \rangle, \emptyset \rangle \\
 r_1 &= \langle V_1, V_2, \text{являются студентами-отличниками}, \text{абстрагирование} \rangle
 \end{aligned}$$

Включение конкретного понятия «студент» в концептуальную модель ПрО требует изменения схемы БД для хранения информации об объектах конкретного понятия. Причем данный вариант БД будет далек от оптимального, однако это является задачей уровня организации данных. Пример схемы БД для рассматриваемого фрагмента модели ПрО до включения конкретного понятия «студент» приведен на рисунке 4, а после включения этого понятия – на рисунке 5.

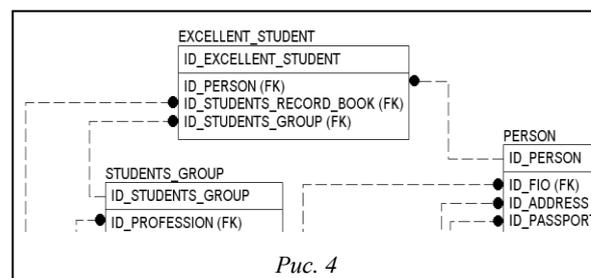


Рис. 4

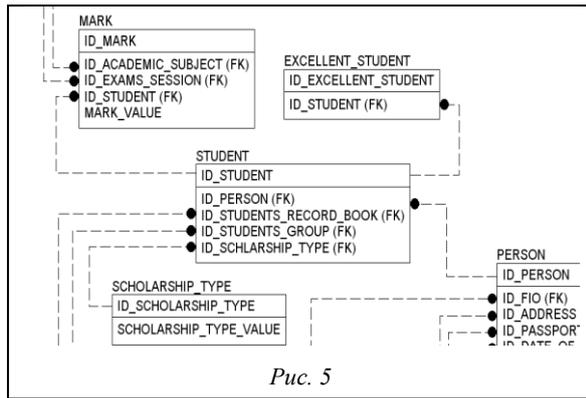


Рис. 5

Модификация системы базовых SQL-запросов будет заключаться в создании SQL-запроса для конкретного понятия и модификации SQL-запроса для понятия-образа.

Формирование базового SQL-запроса для конкретного понятия производится путем выборки всех данных из отношения, содержащего объекты конкретного понятия. Например, базовый SQL-запрос для понятия «студент» представляется в виде

```
CREATE OR REPLACE VIEW v_STUDENT AS
Select *
From STUDENT
```

В общем случае базовый SQL-запрос для конкретного понятия имеет вид:

```
Select *
From R
```

где R – отношение, содержащее объекты конкретного понятия.

Модификация базового SQL-запроса для понятия-образа производится путем естественного соединения отношения, содержащего объекты понятия-образа, с базовым SQL-запросом, формирующим объем конкретного понятия, и проекции на набор свойств, соответствующих понятию-образу. Например, базовый SQL-запрос для понятия «студент-отличник» представляется в виде

```
CREATE OR REPLACE VIEW v_EXCELLENT_STUDENT AS
Select FIO_VALUE, ADDRESS_VALUE, PASSPORT_VALUE, DATE_OF_BIRTH_VALUE, STUDENTS_RECORD_BOOK_VALUE, STUDENTS_GROUP_VALUE, PROFESSION_VALUE, FACULTY_VALUE
From EXCELLENT_STUDENT Natural Join v_STUDENT
```

В общем случае базовый SQL-запрос для понятия-образа имеет вид:

```
Select X1, X2, ..., Xn
From R Join v_V
```

где R – отношение, содержащее объекты понятия-образа; v_V – базовый SQL-запрос, формирующий объем конкретного понятия.

Включение в концептуальную модель ПрО нового конкретного понятия, соответствующего понятию-образу, полученному путем усреднения значений свойств, предполагает включение понятия со значениями, конкретизирующими значения усредненных свойств. Например, определим кон-

кретное понятие «студент» для понятия-образа «успевающий студент». При этом свойство «оценки» у понятия «студент» будут принимать конкретные значения «5», «4», «3» или «2».

Измененный фрагмент модели ПрО формально представляется записью

$$V_1 = \langle \text{успевающий студент}, \langle \text{человек}, \text{зачетка}, \text{группа}, \text{специальность}, \text{факультет} \rangle, \emptyset \rangle$$

$$V_2 = \langle \text{студент}, \langle \text{человек}, \text{зачетка}, \text{группа}, \text{специальность}, \text{факультет}, \text{оценки} \rangle, \emptyset \rangle$$

$$r_1 = \langle V_1, V_2, \text{являются успевающими студентами}, \text{абстрагирование} \rangle$$

Включение конкретного понятия «студент» в концептуальную модель ПрО требует изменения схемы БД для хранения информации об объектах конкретного понятия (фрагмент исходной схемы БД (рис. 2) для случая, когда изначально была создана таблица ADVANCED_STUDENT для хранения объектов понятия «успевающий студент», представлен на рисунке 6; фрагмент схемы БД после включения в модель ПрО конкретного понятия «студент» – на рисунке 7). Причем данный вариант БД будет тоже далек от оптимального, однако это задача уровня организации данных.

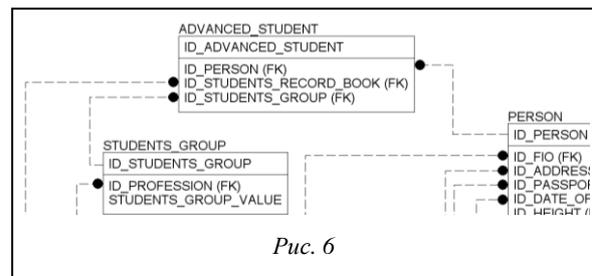


Рис. 6

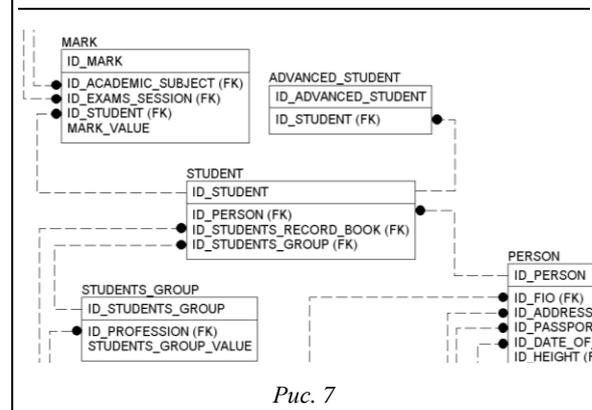


Рис. 7

Изменение системы базовых SQL-запросов будет заключаться в изменении базового SQL-запроса для понятия-образа и в создании базового SQL-запроса для конкретного понятия.

Формирование базового SQL-запроса для конкретного понятия производится путем выборки всех данных из отношения, представляющего объекты конкретного понятия. Например, базовый SQL-запрос для понятия «студент» представляется в виде:

```
CREATE OR REPLACE VIEW v_STUDENT AS
Select *
From STUDENT
```

В общем случае базовый *SQL*-запрос для конкретного понятия имеет вид

```
Select *
From R
```

где **R** – отношение, содержащее объекты конкретного понятия.

Модификация базового *SQL*-запроса для понятия-образа производится путем естественного соединения отношения, хранящего информацию об объектах понятия-образа, с базовым *SQL*-запросом для конкретного понятия и введения предиката, выделяющего требуемые объекты из множества объектов конкретного понятия. Например, *SQL*-запрос для понятия «успевающий студент» представляется в виде

```
CREATE OR REPLACE VIEW v_ADVANCED_STUDENT AS
WITH adv_std AS (Select * From ADVANCED_STUDENT Natural Join v_STUDENT)
Select *
From adv_std
Where exists (select ID_STUDENT, MIN(MARK_VALUE) min_mark, MAX(MARK_VALUE) max_mark from v_MARK where ID_STUDENT = adv_std.ID_STUDENT group by ID_STUDENT having min_mark = 3 and max_mark = 5)
```

В общем случае базовый *SQL*-запрос для понятия-образа имеет вид

```
Select *
From R Join v_V
Where P
```

где **R** – отношение, содержащее объекты понятия-образа; **v_V** – базовый *SQL*-запрос, формирующий объем конкретного понятия; **P** – предикат, формирующий объем понятия-образа.

Рассмотренные варианты применения обобщенной операции абстрагирования позволяют получать принципиально новые представления ПрО, имеющие существенно более высокий уровень абстрагирования и обобщения, чем в существующих ИС. Поддержка отношения абстрагирования делает СИС открытой для формирования специфических типов объектов, обладающих большой выразительной способностью для представления объектов реального мира, а пользователям предоставляет мощные средства для адекватного моделирования ПрО.

Таким образом, реализация отношения абстрагирования делает СИС открытой на уровне концептуального моделирования ПрО.

Литература

1. Чарнецки К., Айзенкер У. Порождающее программирование. Методы, инструменты, применение. СПб: ПИТЕР, 2005. 731 с.
2. Войшвилло Е.К., Дегтярев М.Г. Логика: учеб. для студ. вузов. М.: Изд-во ВЛАДОС-ПРЕСС, 2001. 528 с.
3. Зинченко Р.Е. Системно-изоморфное динамическое соответствие концептуальной модели предметной области и схемы базы данных // Программные продукты и системы. 2010. № 1. С. 71–75.
4. Дрождин В.В., Зинченко Р.Е. Системный подход к концептуальному моделированию предметной области в самоорганизующейся информационной системе // Программные продукты и системы. 2009. № 4. С. 73–79.
5. Том Кайт. Oracle для профессионалов. Кн. 1. Архитектура и основные особенности. М.: «ДиаСофтЮП», 2005. 656 с.
6. Том Кайт. Oracle для профессионалов. Кн. 2. Расширенные возможности и защита. М.: «ДиаСофтЮП», 2005. 816 с.

УДК 658.512

ПРОЦЕССНАЯ МОДЕЛЬ ФОРМИРОВАНИЯ АГРЕГИРОВАННЫХ ТРЕБОВАНИЙ К СЛОЖНЫМ ИНФОРМАЦИОННЫМ СИСТЕМАМ

В.А. Тихомиров, д.т.н.; А.В. Пушина

(Тверской государственный технический университет, tvu@npo-rit.ru);

А.К. Тарасов (НПО «Российские инновационные технологии», г. Тверь)

Рассматривается создание информационных систем для научно-исследовательских, проектных, конструкторских организаций, вузов, ориентированных на поддержку работы с документацией, характерной для всех областей науки и техники. Анализируются способы решения проблем объединения, сжатия и вычленения существенных требований на основании теории нечетких множеств.

Ключевые слова: агрегированные требования, сложные информационные системы.

Создание информационных систем для научно-исследовательских, проектных, конструкторских организаций, вузов, научно-производственных и производственных объединений, ориентированных на поддержку работы с документацией, характерной для всех областей науки и техники, является крайне актуальной проблемой. Это многоэтапный процесс, имеющий, как правило, не

одну стадию. Реализацию его начинают с разработки требований к создаваемой системе любого типа. Наиболее популярными моделями процессов, используемыми в области информационных технологий, являются две – каскадная и спиральная.

В каскадной модели в качестве точек оценки и перехода от одной фазы к другой используются