

А. Н. Родионов, д-р техн. наук, вед. науч. сотр., e-mail: ran@newmail.ru,  
Вычислительный центр ДВО РАН, г. Хабаровск

## Абстрактные роли и примитивы ролевого моделирования сущностей в системе "концептуальная—логическая—физическая модели данных"

*Ролевая концепция достаточно часто используется при построении концептуальных моделей данных. С ее помощью, в частности, появляется возможность динамически выделять подмножества сущностей, которым разрешено участвовать в тех или иных взаимодействиях. Воплощение ролевого поведения на следующем — логическом — уровне моделирования может достигаться либо многократным клонированием сущностных типов по одному клону на каждую роль, либо идти по пути создания специализированных ролевых типов и закрепления последних за определенными типами сущностей. При этом оба подхода порождают сопутствующие проблемы, которые, как в первом случае, приводят к неконтролируемому росту клоновых объектов логических моделей, или, как во втором, требуют использования ролевых констант для обращения к сущностям, играющих определенные роли.*

*Концепция абстрактных ролей, предложенная и реализованная в настоящей работе, устраняет указанные недостатки ролевых типов, ограничиваясь использованием исключительно только той информации, которую содержат логические модели. Разработан универсальный шаблон ролевого моделирования с единственным абстрактным ролевым типом, содержащим список абстрактных ролей. Список постоянен для выбранной конфигурации логической модели. Шаблон включает реальные ролевые типы, каждый из которых привязан к собственному сущностному типу. Такая организация также решает проблему формирования виртуального множества сущностей, играющих одну и ту же роль, но принадлежащих разным сущностным типам.*

**Ключевые слова:** абстрактные роли, ролевое моделирование, ролевые паттерны

### Введение

За последние три десятилетия усилиями многочисленных исследователей все скольконибудь значимые аспекты ролевого моделирования, включая назначение, функции и представление ролей в концептуальных и логических моделях (КМ и ЛМ), были досконально изучены и нашли отражение во множестве публикаций. Проанализировав содержание более чем 20 работ, Ф. Стейманн [1] идентифицировал 15 ролевых характеристик (свойств) и выделил три потенциальных способа представления ролей в моделях данных: посредством именованых отношений, возникающих между объектами [2—4]; придания ролевой семантической окраски объектам, полученным в результате проведения процедур специализации и/или генерализации над множествами, образованными экземплярами исходных объектов [5—7]; простым перечислением экземпляров ролей в специализированных ролевых

объектах и установления соответствия между последними и не ролевыми объектами [8—10].

Ролевая компонента, представленная определенным подмножеством ролевых свойств и использующая один из трех способов представления, присутствует сегодня во всех широко известных и распространенных методологиях моделирования: на основе модели ER (модель "сущность—связь") [4] и ее расширений [7, 11—13]; ORM — объектно-ролевым моделированием [2, 14, 15]; UML — унифицированном языке моделирования [3, 16, 17]; OntoUML — онтологическом UML [18], унифицированной метамодели, интегрирующей ER, ORM и UML [17]. Наблюдаемые различия свидетельствуют об отсутствии консенсуса в вопросах, касающихся прежде всего определения понятия роли, а также возложенных на нее функций. Но даже не этот аспект следует считать определяющим. Восприятие роли зависит от рамок основ, формирующих каркас каждой из методологий. OntoUML, например, строится

на базе развитой системы объектных типов (категорий), включающей непосредственно две ролевые категории: *Role* и *RoleMixin*. В отличие от *OntoUML ORM* организована в терминах объектов, играющих роли, где роль считается обязательным атрибутом любого отношения (взаимодействия, ассоциации).

В настоящей работе также, как и в *OntoUML*, все типы (именованные множества) классифицированы. В первую очередь проводится различие между сущностными и несущностными типами. В первых сосредоточены сведения о сущностях, из которых, собственно говоря, и состоит любая предметная область (ПрО). Вторые являются производными от первых и содержат объекты, выполняющие вспомогательную, обеспечивающую по отношению к сущностям функцию. По мере изложения различные категории несущественных типов будут последовательно появляться, объясняться и описываться в работе.

Постулируется, что роль полностью задает подмножество сущностей, участвующих в том или ином взаимодействии, и является одной из обязательных характеристик взаимодействия. Заметим, что подобная трактовка назначения роли подразумевает закрепление за ней еще одной важной для моделирования данных функции — функции вторичной, динамической подтипизации, обеспечивающей отбор из сущностных типов только тех экземпляров, которые в состоянии играть ту или иную роль. (Аналогичную функцию выполняют и два других известных вида подтипизации — фазная и категорийная подтипизации, которые в работе не рассматриваются.) Укажем, что ролевую подтипизацию от фазной и категорийной отличает то, что "ролевые" множества могут пересекаться.

Базовый, первичный вид подтипизации, обозначенный нами как характеристический, выполняется в начальной фазе концептуального моделирования и формирует каркас модели в виде системы сущностных типов, экземпляры которых не пересекаются. Собственно говоря, наличие такого каркаса, на который вынуждена опираться ролевая подтипизация, является источником одной из ключевых проблем ролевого моделирования, которая проявляется уже на логическом уровне. Обозначенная проблема, в частности, сводится к поиску способа представления неограниченной совокупности ролевых подмножеств в логических моделях, элементами которых к тому же могут быть экземпляры, принадлежащие разным типам.

Другая актуальная задача, которая на первый взгляд может показаться тривиальной, состоит в том, что в случае использования специализированных ролевых таблиц, которые содержат исключительно названия ролей, а не сущности, играющие определенные роли, мощность и содержание таких таблиц становятся известными только в процессе формирования базы данных. Но чтобы обеспечить соблюдение принципа независимости программ и данных, экземпляры ролей должны быть известны уже на этапе проектирования. В противном случае просто невозможно выделить подмножества сущностей, соответствующие той или иной роли. Устранение противоречия между непредсказуемостью и известностью реальных ролей на этапе проектирования, а также решение проблемы смешивания экземпляров, принадлежащих разным типам, но играющих одну и ту же роль, составляют основное содержание настоящей работы. В частности, в статье излагаются решения, основанные на использовании абстрактных ролей, число и состав которых постоянен для любой предметной области.

В название работы вынесено словосочетание "физическая модель", которое призвано подчеркнуть необходимость комплексного решения задач моделирования, не ограниченного рамками только формальных концептуальной и логической моделей. "Физическая" означает, что данных, размещенных в таблицах баз данных и структурированных в соответствии с логической моделью, должно быть достаточно для динамического формирования экземпляров ролевых подмножеств в процессе обращения к этим данным из прикладных программных сред.

## **1. Категории объектных типов предметных областей и концептуальных моделей данных**

Ролевые поведения демонстрируют далеко не все категории типов, которые принято различать в ПрО. Если основываться на том, что ПрО состоит из сущностей, то роли будут характерны только для сущностей. Поэтому вопрос о том, какие типы относить к сущностям, а какие нет — ключевой для ролевого моделирования. Будем опираться на подходы, явно классифицирующие типы в соответствии с их назначением и поведением, которые они демонстрируют в моделируемых универсумах. Ввиду обширности, неоднозначности и многогранности исследований, посвященных различным

аспектам типизации объектов, ограничимся кратким изложением состояния этого вопроса.

Категорийное, шаблонное моделирование сегодня — один из трендов в концептуальном моделировании данных, позволяющий ускорять и упрощать процесс проектирования. Разные исследователи выделяют разные категории объектных типов. Сошлемся на ряд заметных результатов, полученных в этой области. Обширную и развитую систему типов, которая включает такие виды, как *Kinds*, *Subkinds*, *Roles*, *RolesMixins*, *Phases*, *Categories* и *Mixins*, обосновал и предложил в рамках унифицированной фундаментальной онтологии (UFO) Ж. Гуиззарди [18], она получила дальнейшее развитие в лице MLT — многоуровневой теории [19]. Все перечисленные типы для их индустриальной поддержки в последующем в виде профайла были встроены в UML, а также использованы для разработки многочисленных шаблонов концептуального моделирования.

Различные категории типов обнаруживаются и в другой популярной методологии — ORM [2], несмотря на то что на их обособлении не делается явного акцента. В отдельные категории сведены нелексические (сущности — *Entities*) и лексические (значения — *Values*) типы, некоторые ассоциации, трактуемые как самостоятельные типы (*Objectified Associations*), типы категоризации (*Categorization Types*), коллекционные типы (*Collection Types*). Аналоги "*Objectified Associations*" присутствуют и в работе [18], но там они получили более компактное название — "*Relators*", которого и будем придерживаться далее.

В работе [20] показано, что для комплексного моделирования и представления сущностей, принадлежащих одному и тому же типу, требуется в общем случае сформировать сущностный кластер, представленный такими взаимосвязанными сущностными подтипами, как *Patterns*, *Samples* и *Instances*.

По сути, каждая категория (класс) типов демонстрирует собственное, отличное от других категорий поведение. Но все авторы упомянутых работ выделяют сущностные типы в качестве базовых типов. Дать исчерпывающее определение сущности и сформировать критерий, на основании которого можно было бы определить — относится ли объект к сущности или нет, оказалось весьма непросто. Данный вопрос до сих пор остается открытым, и многие исследователи стараются обойти его стороной, в том числе ссылаясь либо на очевидность или тождественность объектов и

сущностей, либо используя общие формулировки наподобие следующих: сущность есть "некоторый объект, представляющий интерес для организации" [21]; сущность — это "нечто, принадлежащее объективной реальности, обличенное в материальную форму" [22]; "сущность — это вещь, которая может быть однозначно идентифицирована" [4].

"Сущностный" вопрос — это вопрос, касающийся в большей степени не информатики, а скорее лингвистики и философии. Не претендуя на универсальность и какую-либо общность, выделим два приема практической направленности, которые могут оказать помощь в идентификации сущностных объектов. Например, если мысленно зафиксировать время, то в ПрО останутся только сущности. Останутся процессы, часто воспринимаемые как сущности, исчезнут ролевые и фазовые объекты, не станет семантических объектов и объектов категоризации. Другой способ выделить сущности в массе объектов — последовательно исключать все другие категории до тех пор, пока в остатке не останутся сущности.

Все дальнейшее изложение будет касаться описания поведения и представления на концептуальном, логическом и физическом уровнях моделирования пар "тип сущности—роли".

## 2. Роль как объект моделирования

Роль, если рассматривать ее как лингвистическую категорию, можно соотнести с некоторым набором объектов и двумя подмножествами действий, которые эти объекты в состоянии совершать, воздействуя на другие объекты или на самих себя, либо, напротив, подвергаться действиям со стороны других объектов, в том числе также оказывая воздействия на самих себя. "Пилот", "врач", "преподаватель", "военнослужащий" — примеры ролей, соотнесенных с объектами такого типа, как "личности".

Представленная ролевая трактовка влечет за собой необходимость в дополнительной конкретизации отношений, возникающих как между сущностями в ПрО, так и между сущностями и другими видами объектов. (Будем в дальнейшем ссылаться на такие взаимодействия как на ролевые и, соответственно, использовать прилагательное "ролевой" для конкретизации отдельных элементов: экземпляров типов, самих типов и свойств экземпляров, имеющих отношение к взаимодействиям подобного рода.) Помимо действий, выполняе-

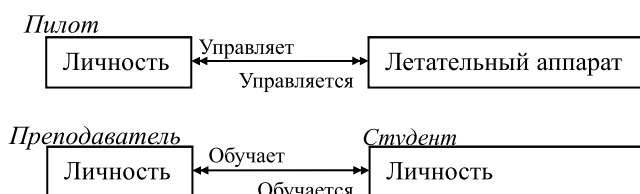


Рис. 1. Пример ассоциаций концептуальной модели, содержащих ссылки на роли

мых сущностями в ассоциации, потребуется указывать и их роли. Для этого ссылку на роль достаточно "прикрепить" к соответствующему типу сущности (рис. 1), экземплярам которой разрешено вступать в то или иное взаимодействие с экземплярами других типов ПрО.

Справедливости ради отметим, что это не единственный способ представления ролей на концептуальном уровне. Д. Кабот и Р. Равентос в работе [23] приводят и описывают еще четыре потенциальных варианта. Но все эти варианты в той или иной степени предполагают задействование произвольного числа ролевых типов, что идет вразрез с поставленной ранее задачей ограничения их числа.

Очевидно, что конкретные роли в состоянии играть только определенные подмножества тех же "личностей", а элементы, входящие в состав этих подмножеств, должны обладать какими-то уникальными характеристиками. Например, чтобы быть "пилотом" или "преподавателем", нужно соответствовать отличительным квалификационным требованиям.

Из сказанного следует, что роль одновременно задает и круг объектов, обладающих определенными характеристиками, и список действий, выполняемых этими объектами

(или перечень действий, направленных на эти объекты), и спектр отношений, в которых "ролевые" объекты могут принимать участие.

Таким образом, чтобы формально задать (описать) некоторую роль, достаточно:

- указать отношения, к которым она будет прикреплена;
- перечислить типы сущностей и в каждом типе выделить подмножество экземпляров, которые в состоянии играть эту роль;
- каждой паре <тип сущности, тип отношения> поставить в соответствие или множество действий, которые выполняют сущности в этом отношении, или множество действий, которым они подвергаются. (Последний аспект ввиду его очевидности и однозначности вынесен за рамки обсуждения.)

Отношения между перечисленными категориями, имеющими непосредственное отношение к роли, представлены на рис. 2. Здесь:  $R$  — множество ролей;  $E$  — множество сущностей;  $C$  — множество отношений, один из элементов которых обязательно тип сущности;  $E_{r_z}^n$  —  $n$ -е подмножество сущностей, которые потенциально могут играть  $r_z$ -ю роль (Одну и ту же роль могут выполнять объекты разной природы, сгруппированные по этой причине в отдельные, непересекающиеся друг с другом подмножества  $E_{r_z}^n$ );  $\bar{E}_{r_z}^n$  — собственное подмножество  $E_{r_z}^n$ , элементы которого в состоянии играть роль,  $\bar{E}_{r_z}^n \subseteq E_{r_z}^n$ . (Об отличиях  $E_{r_z}^n$  и  $\bar{E}_{r_z}^n$  — чуть ниже.)

Как было отмечено ранее, важнейшая функция роли, вытекающая из ее определения, — выделение в ПрО именованных подмножеств сущностей, которые способны играть эту роль. (Способны, однако, не означает, что сущность действительно будет участвовать в ролевых взаимодействиях.

Необходимо еще, чтобы роль была каким-то образом прикреплена к сущности.) Априорное именование ролевых подмножеств, являющееся обычной практикой в ПрО, дает основание предположить, что такие подмножества не пересекаются. Приведем два примера, взятых из практики. Роль — "объекты транспортного налога" — различает автомобили, мотосани, моторные лодки, гидроциклы и другие разновидности сущностных объектов. Другая роль "Экзаменаторы" может выполняться только или доцентами или профес-

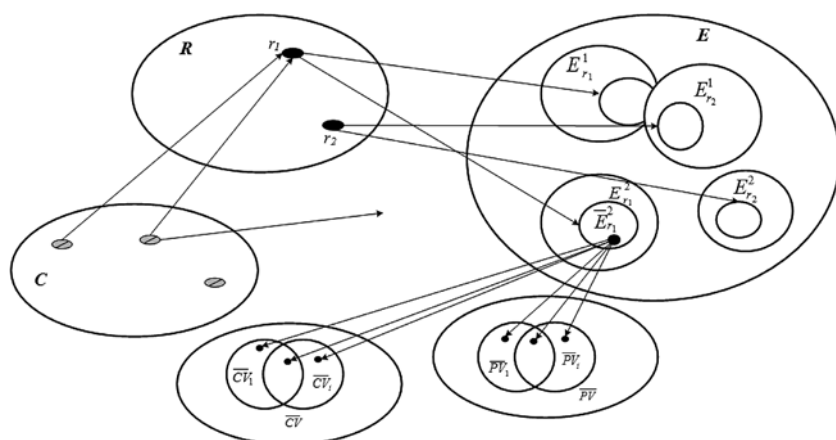


Рис. 2. Соответствия между элементами актуальных множеств, участвующих в ролевых взаимодействиях

сорами. Таким образом, для  $\forall r_z$  справедливо:

$$\bigcap_{n=1 \div N} E_{r_z}^n = \emptyset.$$

В то же время стандартной является ситуация, когда сущности, принадлежащие одному типу, играют множество ролей, и поэтому их отдельные подмножества могут пересекаться. Те же моторные лодки — одновременно и объекты налогообложения, и средства передвижения.

Для многих ролей одной только ссылки на  $E_{r_z}^n$  оказывается недостаточно, чтобы очертить круг "ролевых" сущностей (сущностей, задействованных во взаимодействиях, к которым прикреплены роли), потому что не все элементы  $E_{r_z}^n$  в состоянии играть  $r_z$ -ю роль в конкретный момент времени. Для того чтобы отдельные экземпляры  $E_{r_z}^n$  могли участвовать в подобных отношениях ("ролевых" отношениях), часть подлинных свойств этих экземпляров ("ролевых" свойств) должна иметь на этот момент времени вполне определенные значения, и/или они должны участвовать в каких-то взаимодействиях, а экземпляры этих взаимодействий, в свою очередь, также должны принимать наперед заданные значения. Например, мощность моторных лодок как объекта налогообложения должна превышать 5 лошадиных сил. Для занятия должности "профессор" необходимо иметь степень доктора наук и/или ученое звание не ниже доцента. В последнем случае — это уже экземпляры отношений, возникающих между экземплярами сущности "Личности" и объектами квалификационных типов "Ученые степени" и "Ученые звания" соответственно. (Упомянутые выше подлинные свойства — это свойства, которые, согласно работе [24], присущи экземплярам типа, изолированного от других типов.)

В связи с этим о  $\bar{E}_{r_z}^n$  можно говорить как о динамическом множестве, состав которого может меняться в моменты времени  $t$ , когда происходят изменения в значениях "ролевых" свойств и/или экземплярах "ролевых" отношений.

Представим то же самое формально. Требуется задать множества  $\bar{E}_{r_z}^n(t)$ . Запись  $\bar{E}_{r_z}^n(t)$  будет означать, что состав  $\bar{E}_{r_z}^n$  фиксируется на момент времени  $t$ . Будем также полагать, что содержание  $E_{r_z}^n$  априорно известно. По крайней мере, в ПрО не только принято именовать все  $E_{r_z}^n$ , но и предоставлять описание элементов  $E_{r_z}^n$ . Обозначим  $p_m$  —  $m$ -е подлинное свойство сущности  $e_{r_z}^n \in E_{r_z}^n$ , а  $v_{p_m}(t)$  — значение этого свойства в момент времени  $t$ . Пусть  $\overline{PV}$  — множество значений подлинных свойств, которые задают  $r_z$ -ю роль для элементов множества  $E_{r_z}^n$ .

Другими словами,  $\overline{PV}$  — это множество индикаторов, по достижении которых  $e_{r_z}^n$  переходит в категорию  $\bar{e}_{r_z}^n$ . В общем случае, для  $\forall t$  возможны ситуации, когда  $PV(t) \not\subseteq \overline{PV}$ , т. е. никакие элементы  $\bar{e}_{r_z}^n$  не в состоянии играть роль  $r_z$ . В простейшем варианте

$$\overline{PV} = \{ \langle p_m, [\bar{v}_{p_m}^s, \bar{v}_{p_m}^q], T_{p_m} \rangle \}_1^M, \quad (1)$$

где  $M$  — число подлинных свойств сущностей, образующих  $E_{r_z}^n$ ;  $[\bar{v}_{p_m}^s, \bar{v}_{p_m}^q]$  — диапазон значений, которое должно приобрести свойство, чтобы выполнять роль;  $\bar{v}_{p_m}^s$  и  $\bar{v}_{p_m}^q$  — соответственно начальное и конечное значения свойства;  $T_{p_m}$  — имя домена, на котором определено свойство. (Далее  $T_{p_m}$  как несущественное для обсуждения будем опускать.) Так как не все свойства задают ролевую принадлежность сущностей, а индекс  $m$  в (1) "пробегают" по всем свойствам, исключим "неролевые" свойства, присвоив  $\bar{v}_{p_m}^s$  и  $\bar{v}_{p_m}^q$  значение *null* в соответствующем триplete  $\langle p_m, [\bar{v}_{p_m}^s, \bar{v}_{p_m}^q], T_{p_m} \rangle$ .

Приведем пример задания  $\overline{PV}$  для множества тех же моторных лодок.  $\overline{PV} = \{ \langle \text{Мощность}, 5 \rangle, \langle \text{Грузоподъемность}, 700 \rangle \}$ .

Как показывают наблюдения, практика в отношении  $\overline{PV}$  не ограничивается безальтернативностью. В реальности таких множеств, как  $\overline{PV}$ , может оказаться несколько. Например:

$$\overline{PV}_1 = \{ \langle \text{Мощность}, 5 \rangle, \langle \text{Грузоподъемность}, 700 \rangle \}$$

$$\overline{PV}_2 = \{ \langle \text{Мощность}, 9 \rangle, \langle \text{Грузоподъемность}, 700 \rangle, \langle \text{Материал}, \text{ПВХ} \rangle \}.$$

Поэтому в действительности  $\overline{PV}$  — это система из  $l$ -множеств ( $l = \overline{1}, \overline{L}$ ), задающая совокупность ролевых альтернатив:

$$\overline{PV} = \{ \overline{PV}_1, \overline{PV}_2, \dots, \overline{PV}_l, \dots, \overline{PV}_L \}.$$

В выражении (1)  $\bar{v}_{p_m}^s$  и  $\bar{v}_{p_m}^q$  — это константы. В то же время не исключены ситуации, когда и  $\bar{v}_{p_m}^s$ , и  $\bar{v}_{p_m}^q$  задаются посредством формул. Поэтому более корректно использовать предикатную форму для задания  $\bar{v}_{p_m}^s$  и  $\bar{v}_{p_m}^q$ .

По аналогии с  $\overline{PV}_l$  обозначим через  $PV_l^{e_{r_z}^n}(t)$   $l$ -е множество значений подлинных свойств элемента  $e_{r_z}^n$ , которые он приобретает в момент времени  $t$  (ввиду громоздкости верхнего индекса в  $PV_l^{e_{r_z}^n}(t)$  последний будем опу-

скать, полагая, что  $PV_f(t)$  по умолчанию ссылается на  $e_{r_z}^n$ :

$$PV(t) = \{ \langle p_m, v_{p_m}(t) \rangle \}_1^M,$$

где  $v_{p_m}(t)$  — значение  $p_m$ -го свойства  $e_{r_z}^n$  в момент времени  $t$ .

Ориентируясь на наперед заданные индикаторы в виде  $\overline{PV}$ , состав  $\overline{E}_{r_z}^n(t)$  можно записать как:

$$\begin{aligned} \overline{E}_{r_z}^n(t) = & \left\{ e_{r_z}^n \mid \left( PV(t) \xrightarrow{f_b} \overline{PV}_1 \right) \vee \dots \right. \\ & \left. \dots \vee \left( PV(t) \xrightarrow{f_b} \overline{PV}_l \right) \vee \dots \vee \left( PV(t) \xrightarrow{f_b} \overline{PV}_L \right) \right\}. \end{aligned} \quad (2)$$

Дадим пояснения касательно  $PV(t) \xrightarrow{f_b} \overline{PV}_l$ . Из постановки задачи вытекает, что между двумя множествами должно быть установлено взаимно-однозначное соответствие. Индекс 'b' в обозначении  $f_b$  говорит о том, что  $f_b$  — биекция. Согласно определению биекции каждому  $\langle p_m, v_{p_m}(t) \rangle$  должен соответствовать  $\langle p_m, [\overline{v}_{p_m}^s, \overline{v}_{p_m}^q] \rangle$ . Для этого необходимо, чтобы  $v_{p_m}(t) \in [\overline{v}_{p_m}^s, \overline{v}_{p_m}^q]$ .

Аналогичным образом формализуем второе условие принадлежности  $e_{r_z}^n$  к  $\overline{E}_{r_z}^n$ , также имеющее непосредственное отношение к участию сущности в предопределенных взаимодействиях. Как и в случае со свойствами, сформируем в первую очередь систему множеств  $\overline{CV}$ , определяющих индикативные экземпляры взаимодействий:

$$\overline{CV} = \{ \overline{CV}_1, \overline{CV}_2, \dots, \overline{CV}_c, \dots, \overline{CV}_C \},$$

где  $\overline{CV}_c$  — множество экземпляров взаимодействий, в которых могут участвовать экземпляры "ролевых" сущностных типов.

В отличие от свойств, которые отличает простой формат представления  $\langle \text{Имя свойства} - p_m, \text{Значение свойства} - v_{p_m} \rangle$ , для взаимодействий таких форматов может быть предложено несколько. Возьмем за основу очевидный и одновременно универсальный формат, используемый для представления типов, отнесенных к категории "Слабые сущности". При этом будем руководствоваться следующими соображениями.

1. В общем случае предельная кардинальность, которая может быть достигнута во взаимодействии "ролевых" сущностных типов с другими категориями объектов, — M:M —

многие ко многим. "Слабые сущности" — один из двух допустимых типов структур (наряду с "документальными структурами" [25]), который подходит для хранения результатов подобных взаимодействий. В том числе и взаимодействий с кардинальностью 1:1, 1:M, M:1.

2. "Слабые сущности" способны отражать взаимодействия двух и более категорий объектов типов. В демонстрационных целях ограничимся двумя категориями, одна из которых — "ролевой" сущностный тип, другая — тип, с которым взаимодействует ролевой тип.

3. Результатом взаимодействия, кроме факта самого взаимодействия, может стать совокупность значений, приобретаемых какими-то характеристиками, которые, в противовес подлинным свойствам, трактуются как взаимные свойства [23]. Например, претендент на должность должен иметь квалификацию "преподаватель", стаж работы преподавателем не менее 5 лет и опубликовать не менее трех научных работ за последние четыре года.

Таким образом, с учетом сказанного, множество  $\overline{CV}_c$  можно определить следующим образом:

$$\overline{CV}_c = \{ \langle We_w, \{ \langle Id_h, \overline{V} \rangle \}_1^H \rangle \}_1^W,$$

где  $We_w$  —  $w$ -я "слабая сущность";  $Id_h$  — идентификатор  $h$ -го объекта, с которым взаимодействует экземпляр ролевого типа;  $\overline{V}$  — множество значений характеристик взаимодействий.

В свою очередь, совокупность взаимодействий и результатов этих взаимодействий, в которых сущность состоит в момент времени  $t$ , найдется как

$$CV(t) = \{ \langle We_w, \{ \langle Id_h, V \rangle \}_1^H \rangle \}_1^W.$$

Далее, по аналогии с (2), запишем второе условие принадлежности  $e_{r_z}^n$  к  $\overline{E}_{r_z}^n$ :

$$\begin{aligned} \overline{E}_{r_z}^n(t) = & \left\{ e_{r_z}^n \mid \left( CV(t) \xrightarrow{f_b} \overline{CV}_1 \right) \vee \dots \right. \\ & \left. \dots \vee \left( CV(t) \xrightarrow{f_b} \overline{CV}_c \right) \vee \dots \vee \left( CV(t) \xrightarrow{f_b} \overline{CV}_C \right) \right\}. \end{aligned} \quad (3)$$

Выражения (2) и (3) справедливы, если принадлежность  $e_{r_z}^n$  к ролевому подмножеству находится на основании либо совпадения ролевых свойств  $\left( PV(t) \xrightarrow{f_b} \overline{PV}_l \right)$ , либо ролевых от-

ношений  $\left( CV(t) \xrightarrow{f_b} \overline{CV}_c \right)$ . Но нередко вопрос о  $e_{r_z}^n \in \overline{E}_{r_z}^n$  решается с учетом выполнения и первого, и второго условий.

Если бы отсутствовала альтернативность, то выражение для  $\overline{E}_{r_z}^n(t)$  имело бы простую форму:

$$\overline{E}_{r_z}^n(t) = \left\{ e_{r_z}^n \mid \left( PV(t) \xrightarrow{f_b} \overline{PV} \right) \wedge \left( CV(t) \xrightarrow{f_b} \overline{CV} \right) \right\}.$$

Альтернативность же предполагает, что индикаторную миссию выполняют наперед заданные пары  $(\overline{PV}_l, \overline{CV}_c)_j \in \overline{PV} \times \overline{CV}$ . Индекс  $j$  указывает на элемент прямого произведения множеств  $\overline{PV}$  и  $\overline{CV}$ . Воспользуемся порождающей процедурой для нахождения элементов  $\overline{E}_{r_z}^n$ :

$$\overline{E}_{r_z}^n(t) = \left\{ \begin{array}{l} e_{r_z}^n \mid \left( PV(t) \xrightarrow{f_b} \overline{PV}_l \right) \wedge \left( CV(t) \xrightarrow{f_b} \overline{CV}_c \right) \\ \text{хотя бы для одной } (\overline{PV}_l, \overline{CV}_c)_j \end{array} \right\}.$$

Завершив построение ролевой модели, еще раз обратимся к отношениям, возникающим между элементами множеств, вовлеченных в ролевой контент (рис. 2), и зададимся следующими вопросами: а не чрезмерно ли усложнена такая система? В какой степени она объективна и отражает естественное положение вещей, а в какой субъективна — задумана исключительно для проектирования баз данных, подчиняющихся априори сформулированным принципам их организации?

Если обратить пристальное внимание на любой из распространенных в мире языков человеческого общения, то во всех них обнаружится большое число существительных с "ролевым" лексическим значением. И в этом плане "ролевые" существительные, будучи продуктом человеческого сознания, — такая же объективная реальность, как и сами сущности.

Если взглянуть на систему ролевых отношений (рис. 2) с точки зрения эффективности организации баз данных, то можно обнаружить транзитивную зависимость, "проходящую" через множество ролей, и исключить эту зависимость. Правда, в этом случае пришлось бы полностью отказаться от богатой семантики, сосредоточенной в ролевых экземплярах, которая делает осмысленной последующее использование ролевых подмножеств в приложениях баз данных.

### 3. Представление ролей на логическом уровне.

#### Ролевая подтипизация

До сих пор ролевые объекты никоим образом явно не обозначали себя на концептуальном уровне. Роли присутствовали в КМ неявно в виде ссылок на именованные подмножества объектов, которые в состоянии играть ту или иную роль. И этого было вполне достаточно, чтобы представить ролевую составляющую связи между объектами.

Включение ролей в концептуальные модели автоматически влечет за собой решение вопроса о способе их представления в логических моделях. Сделать это можно несколькими способами. Рассмотрим все известные способы, выделив положительные и отрицательные стороны каждого решения.

Но первое, с чем следует определиться и от чего далее будем отталкиваться — каркас будущей логической модели, образуемый типами сущности. Как было отмечено ранее, начальный этап моделирования данных всегда предполагает разбиение множества сущностей Про на непересекающиеся подмножества, именуемые типами. (Чтобы различать другие виды "разбиений", будем именовать подобное разбиение характеристическим, или характеристической подтипизацией (ХП).) Основанием для проведения ХП выступает необходимость в обособлении групп однородных сущностей, отличающихся одинаковыми наборами подлинных свойств. Тем самым достигается неразрезанность таблиц (по одной для каждой группы), в которых предполагается размещать экземпляры этих групп. Заметим, что процедура разбиения носит эвристический характер, и типы сущностей как объекты логических моделей появляются уже на концептуальном уровне. Будем опираться на каркас, образованный типами сущностей, полученными в процессе выполнения ХП, и полагать его отправной точкой для последующего ролевого моделирования.

На основании полученного ранее формального описания роли сформулируем основные требования, которым должны удовлетворять логические структурные подсхемы, инкапсулирующие сущности и роли.

Подсхемы должны:

- обеспечивать смешивание экземпляров различных типов сущностей при реализации ими одинаковых ролей (канонический пример — роль покупателя, которую могут одновременно играть и личности, и организации, если экземпляры личностей и орга-

низаций относятся к разным типам) (требование 1);

- быть в состоянии бесппроблемно (не нарушая принцип независимости программ и данных) выделять в логической и физической моделях ролевые подмножества сущностей, участвующие во взаимодействиях. Более конкретно — запросы к базам данных должны манипулировать исключительно названиями таблиц и не ссылаться на записи таблиц (требование 2).

Если руководствоваться изложенными выше соображениями о назначении и функции роли, то принципиальных решений по "ролевому" логическому моделированию может быть только два (рис. 3). Первое предполагает использование специализированного ролевого типа, содержащего список ролей. Второе — создание типов (по одному типу — "ролевому" сущностному типу — на каждую роль), в которых перечисляются сущности, играющие ту или иную роль. Несмотря на то что последнее решение не создает трудностей в выполнении требования 2, оно не может считаться полностью приемлемым, так как его использование ведет к необоснованному дублированию данных в "ролевых" сущностных типах. (Напомним, одна и та же сущность может играть множество ролей и поэтому будет одновременно присутствовать в качестве экземпляров нескольких типов.) Другой серьезный изъян — вследствие множественности и непостоянного числа ролей размерность модели данных, с одной стороны, будет превышать любые разумные пределы, а с другой, — состав модели будет периодически пересматриваться всякий раз, когда будет появляться очередная роль.

Указанных недостатков лишены две подсистемы, задействующие ролевые типы (рис. 3, б). Правда, и они не обеспечивают выполнение тре-

бования 1. Обратимся к стандартному варианту запроса (см. ниже *Листинг 1*) на выборку сущностей, играющих конкретную роль. Сделаем это на примере типа сущности "Личности (Persons)". Запрос ссылается на ролевой тип "Roles".

*Листинг 1*

```
SELECT Persons.Surname, Persons.Name, Persons.Patronymic
FROM Roles INNER JOIN Persons_Roles ON Roles.Id_role = Persons_Roles.Id_role LEFT OUTER JOIN Persons ON Persons_Roles.Id_prs = Persons.Id_prc
WHERE (Roles.Id_role = 2)
```

Такой запрос в предложении WHERE всегда будет содержать явный указатель (в нашем примере 2) на конкретную роль (Roles.Id\_role = 2). Но при этом следует иметь в виду, что таблица ролей в процессе эксплуатации базы данных заполняется произвольным образом. Поэтому на этапе проектирования логической модели неизвестно, какой идентификатор в базе данных будет присвоен конкретной роли и каким будет сам ролевой список — сколько и каких элементов он будет включать.

Справедливости ради следует заметить, что литерал '2' можно было бы заменить переменной и присваивать ей значение в программном коде, предусмотрев некоторый интерактив, во время которого пользователь мог бы выбрать нужную роль из соответствующего списка. Но подобный выход, что очевидно, явно усложнит работу с приложением. Тем не менее, какой-либо другой альтернативы, кроме как использовать ролевые типы, не просматривается. (По крайней мере, в системе тех канонических понятий, принципов и категорий моделирования, которые сегодня применяются для организации баз данных.) Исходя из этого решение проблем ролевого моделирования будем искать в плоскости трансформации схемы (рис. 3, б), что и будет сделано далее.

Как развитие схемы с ролевым типом на рис. 3, б приведен еще один вариант, в котором ролевые типы непосредственно привязаны к сущностным типам. Несмотря на то что при такой организации также не выполняется требование 1 и частично требование 2 (для его выполнения потребуется продублировать уже экземпляры ролей), примем за основу данную схему, потому что, как показывает практика и что будет продемонстрировано ниже, наблюдаются заметные различия в процедурах, связанных с формированием экземпляров ролей, "играемых" различными сущностными типами.

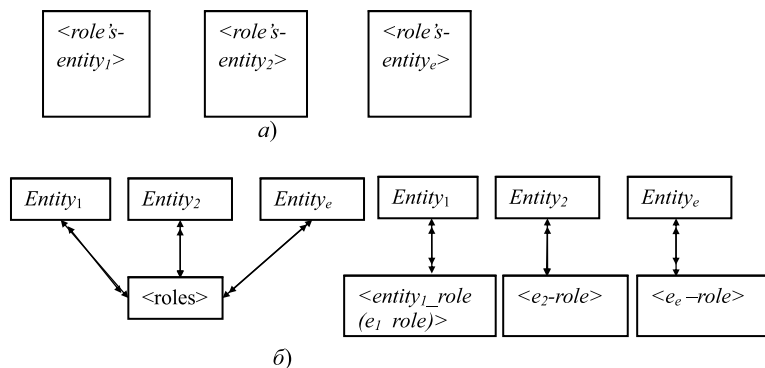


Рис. 3. Способы представления ролей в логических моделях:

а — с "ролевыми" сущностными типами; б — с ролевыми типами



Вернемся к исходному разбиению множества сущностей ПрО, задающему каркас и концептуальной, и логической моделей данных в виде совокупности непересекающихся типов. Не будем останавливаться на возможных вариантах подобных разбиений — их может быть неограниченное число, и это не предмет настоящего исследования. Заметим, что роли выполняют фактически схожую функцию — формируют те же множества сущностей, но только на основании другого признака — участия сущностей в определенных взаимодействиях. Что это дает?

Ролевая подтипизация — это, по сути, один из универсальных адаптационных инструментов моделирования данных, позволяющий получить для любого состояния предметной области ее адекватную модель, в том числе исправить ошибки или неточности, допущенные во время проведения характеристической подтипизации. Например, типы сущностей "Студенты" и "Преподаватели" без труда приводятся к типу "Личности" простой заменой <тип "Личности"> на <роль "Личности">.

#### 4. Абстрактные роли

Как было уже сказано, решение задач ролевого моделирования следует искать в плоскости трансформации схемы (рис. 3, б) с явно выделенными ролевыми типами — по одному ролевому типу на каждый сущностный тип. Обратим внимание на то, что и концептуальный, и логический уровни моделирования не в состоянии оперировать экземплярами *реальных* ролей, которые появляются только по мере заполнения базы данных конкретной предметной области. Следовательно, роли и в КМ, и в ЛМ могут быть только *абстрактными*. Поставим в соответствие, пока только на концептуальном уровне, каждой компоненте отношения по одной абстрактной роли — так, как это было сделано на рис. 1. Заметим, что в этом случае абстрактные роли становятся такими же полноправными объектами КМ, как и другие объекты. Полноправными означает, что их число в КМ — число постоянное, и им всем присвоены уникальные имена.

Следующая задача — каким образом абстрактные роли представить в ЛМ и как связать их с ролевыми типами, в которых будут перечислены реальные роли? Сделать это можно достаточно просто, если разместить список абстрактных ролей в специализированном типе "*Abstract\_Roles — Абстрактные роли*".

Фактически, абстрактные роли — это такие же константы ЛМ, каковыми являются таблицы и отношения: их число известно, и им всем присвоены уникальные имена. Следовательно, не возникает никаких сложностей при обращении ко всем ним из программного кода.

Включение в состав логической модели таких ролевых констант несколько нарушает каноническое представление о видах ее компонентов, но в то же время постоянство состава абстрактных ролей, вытекающее из постоянства отношений логической модели, дает основание для подобного расширения логической модели, не вступая в какие-либо противоречия с принципами организации баз данных. Таким образом, к классическим объектам логических моделей — таблицам и ассоциациям — добавляется конечное множество априори идентифицированных записей таблицы "*Абстрактные роли*". Связав сущностные таблицы с их ролевыми таблицами и установив связи между ролевой таблицей и таблицей "*Абстрактные роли*", придем к схеме, показанной на рис. 4.

(Ввиду того, что, во-первых, в ЛМ появляются константы и, во-вторых, не указывается вид ЛМ, термины "таблица" и "тип" используются как синонимы, а структура ЛМ описывается в категориях "таблица", "атрибуты", "отношение" (ассоциация) и запись.)

Удостоверимся в том, что полученная конфигурация действительно позволяет одновременно и объединять (смешивать) сущностные экземпляры, принадлежащие разным типам, когда они играют одинаковые роли, и выделять ролевые подмножества, не прибегая к ссылкам на реальные роли (как это было сделано в листинге 1) при обращениях к данным. Для иллюстрации воспользуемся представлением данных, отражающих факты приобретения то-

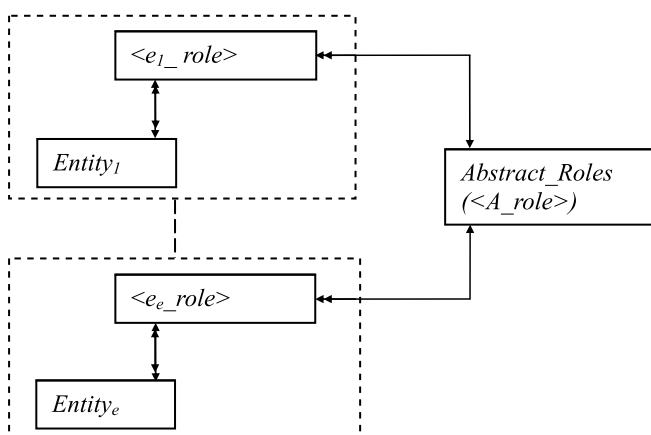


Рис. 4. *Abstract\_Roles* в логической модели

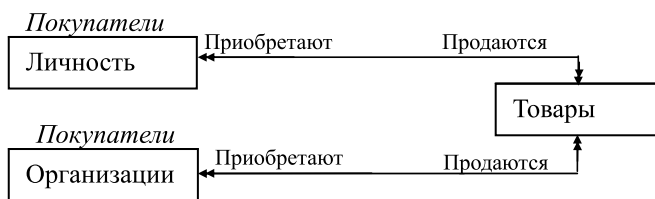


Рис. 5. Концептуальная модель продажи товаров физическим (личности) и юридическим (организации) лицам

варов в некоторой условной предметной области. Концептуальная модель и соответствующая ей логическая модель с фрагментами базы данных изображены на рис. 5 и 6.

Сведения о приобретенных товарах (рис. 6) сосредоточены в таблицах "Purchases\_by\_Persons" и "Purchases\_by\_Organizations", отнесенных к категории "слабые сущности — <we>". Ключевые атрибуты помечены символом "\*".

Для выборки подмножества личностей, которые в состоянии играть роль преподавателя, воспользуемся запросом (см. ниже Листинг 2), в котором задействована ролевая константа  $Id_{ar} = 2$ , соответствующая роли "Преподаватель" и размещенная в таблице "Abstract\_Roles".

Листинг 2

```

SELECT Person.Sername, Person.Id_prs
FROM Abstract_Roles INNER JOIN
Roles_of_Persons ON Abstract_Roles.Id_ar = Roles_of_Persons.Id_ar INNER JOIN
Persons_Roles ON Roles_of_Persons.Id_RoP = Persons_Roles.Id_RoP INNER JOIN
Person ON Persons_Roles.Id_prs = Person.Id_prs
WHERE (Abstract_Roles.Id_ar = 2)

```

Итоговая выборка будет содержать список лиц (табл. 1), которые являются преподавателями.

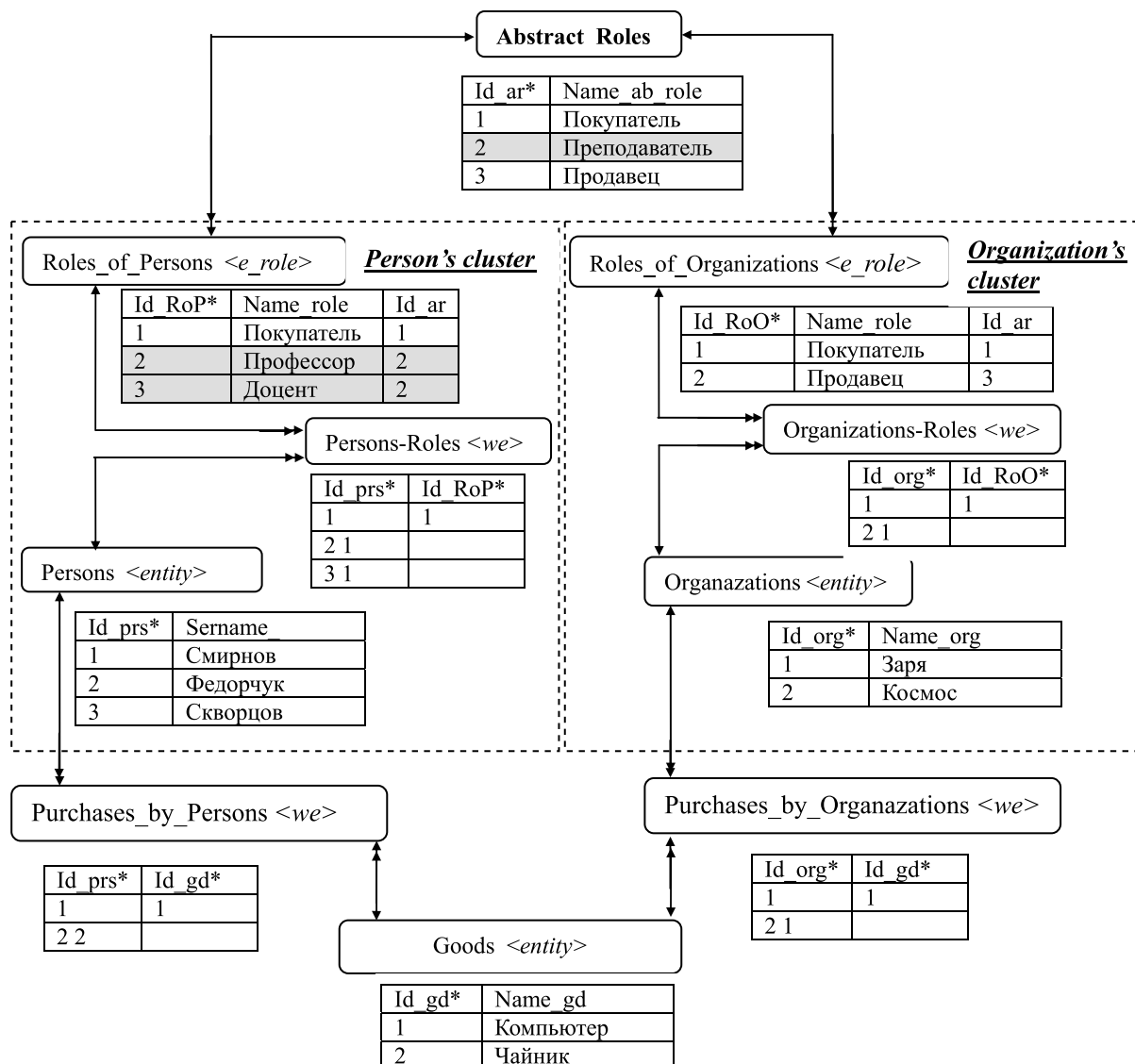


Рис. 6. Фрагмент логической модели с ролевыми объектами и записями базы данных

Таблица 1

Список преподавателей

	Sername	Id_prs
1	Смирнов	1
2	Федорчук	2

Таблица 2

Список покупателей

	Name	Name_gd
1	Смирнов	Компьютер
2	Федорчук	Чайник
3	Заря	Компьютер
4	Космос	Компьютер

Сконфигурировать запрос касательно сущностей, отнесенных к разным типам, но играющих одновременно одну и ту же роль — "покупатель", также не составляет сложностей. Листинг 3 содержит текст SQL-запроса, в результате выполнения которого формируется совмещенный список личностей и организаций, купивших товары (табл. 2).

Листинг 3

```

SELECT Organizatons.Name_org as Name, Goods.
Name_gd
FROM Abstract_Roles INNER JOIN
Roles_of_Organizations ON Abstract_Roles.Id_ar = Roles_
of_Organizations.Id_ar INNER JOIN [Organizatons-Roles]
ON Roles_of_Organizations.Id_RoO = [Organizatons-
Roles].Id_RpO INNER JOIN Organizatons ON [Organizatons-
Roles].Id_org = Organizatons.Id_org INNER JOIN
Purchases_by_Organizatons ON [Organizatons-Roles].
Id_org = Purchases_by_Organizatons.Id_org INNER
JOIN Goods ON Purchases_by_Organizatons.Id_gd =
Goods.Id_gd
WHERE (Abstract_Roles.Id_ar = 1)
Union
SELECT Person.Sername as Name, Goods.Name_gd
FROM Persons_Roles INNER JOIN Person ON Persons_
Roles.Id_prs = Person.Id_prs INNER JOIN Abstract_
Roles INNER JOIN Roles_of_Persons ON Abstract_Roles.
Id_ar = Roles_of_Persons.Id_ar ON Persons_Roles.Id_role
= Roles_of_Persons.Id_RoP INNER JOIN Purchases_
by_Persons ON Persons_Roles.Id_prs = Purchases_by_
Persons.Id_prs INNER JOIN Goods ON Purchases_by_
Persons.Id_gd = Goods.Id_gd
WHERE (Abstract_Roles.Id_ar = 1)

```

## 5. Типы, конкретизирующие роли

До сих пор вопросы, касающиеся состава условных *e\_role*-типов и назначения сущно-

стей на роли были вынесены за рамки обсуждения. Между тем *e\_role*-типы — это такие же абстракции, как и *A\_role*-тип. В "чистом" виде такие типы в ПрО не присутствуют. Их функцию выполняют другие типы.

На примере сущностного типа "Личности", отличающегося развитой и многогранной системой ролевых поведений (о чем свидетельствует богатая лексика, касающаяся обозначений ролей, присутствующая во всех национальных языках), продемонстрируем один из реальных, а не абстрактных *Real\_roles*-типов, и рассмотрим круг вопросов, которые возникают в процессе задания ролей и закрепления личностей за ролями.

Типичный *Real\_roles*-тип — "Должности". На первый взгляд может показаться, что простого перечисления должностей достаточно, чтобы однозначно конкретизировать роли, играемые личностями в ПрО. Но реальность несколько отличается от подобного упрощенного представления. Нередко роль задается объектами нескольких типов. Например, это может быть пара <"Должности", "Подразделения"> (экземпляр — "Инженер 1-й категории", "отдела IT") или тройка типов <"Должности", "Подразделения", "Совместительство">.

В связи с этим уместно ввести некоторый шаблон ролевого представления, который будет включать обязательный определяющий ролевой тип (<*role*>-объект) и произвольное число уточняющих (необязательных) типов (рис. 7).

На показанном шаблоне (рис. 7) символом 'M' (*Mandatory*) обозначен обязательный тип, а символом 'A' (*Adjunct*) — необязательный (вспомогательный). Экземпляры реальных ролей, представляющие собой некоторое подмножество прямого произведения  $M \times A_1 \times \dots \times A_N$ , образуют *Person\_roles*-объект, который уже не является абстракцией. Чтобы отличать ординарные *e\_role*-объекты от реальных ролевых типов, последние на схемах помечаются как *r\_role*-типы.

Поскольку экземпляры "*Persons\_Roles*" участвуют в последующих взаимодействиях (в частности, с "*Persons*" (рис. 6)), уместно каждому экземпляру "*Persons\_Roles*" присвоить уникальный идентификатор и дополнительно классифицировать "*Persons\_Roles*" как <*relator*>. Также следует иметь в виду, что в *relator*-ax, которые являются по сути ассоциациями, следует поддерживать уникальность составного внешнего ключа, образованного из атрибутов, ссылающихся на записи M- и A-объектов. На рис. 7 указанная особенность отражена с помощью

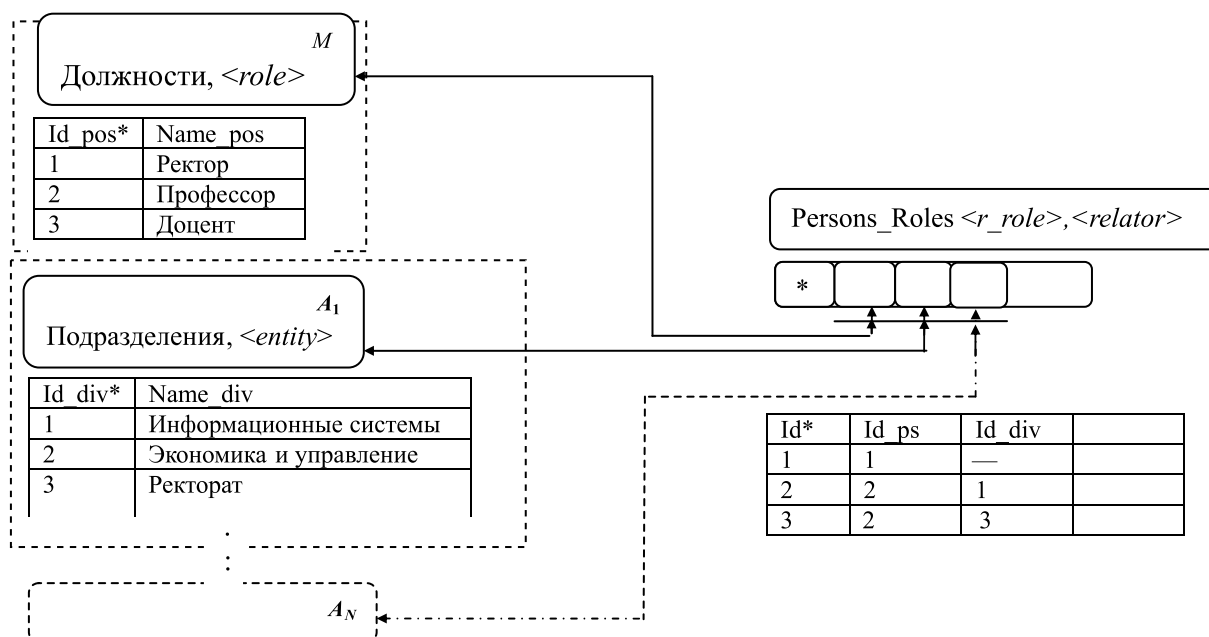


Рис. 7. Шаблон моделирования  $r\_role$ -типов

сплошной черты, охватывающей все соответствующие атрибуты.

Другая актуальная для ролевого моделирования задача — конфигурирование структурных подсем, записи таблиц которых отражают факты закрепления реальной роли за реальной сущностью. Остановимся на следующей постановке данной задачи: для того чтобы сущность была способна играть ту или иную роль, должны совпасть характеристики этой роли и конкретной сущности. То же самое было ранее представлено в работе в виде формальной модели роли.

Применительно к личностям и их ролям такими характеристиками обычно становятся квалификационные требования, возрастные ограничения и другие аналогичные признаки. В предметных областях нетрудно обнаружить объекты, несущие сведения о квалификациях. На примере моделирования квалификационных требований покажем, что это за объекты, и к каким паттернам моделирования приводит включение типов, содержащих подобные объекты, в состав моделей данных.

## 6. Назначение на роли и ограничения в использовании ролевых типов

Из представленного ранее определения роли вытекает, что сущность в состоянии играть ту или иную роль, если она (сущность) обладает определенными качествами (свойствами). Например, чтобы стать участником Олимпий-

ских игр, нужно предварительно получить лицензию на участие. Или, к чтению лекций допускаются только лица, имеющие звание доцента или профессора. Следовательно, множество таких характеристик и значений этих характеристик полностью задают конкретную роль. Отсюда следует тот факт, что для того чтобы закрепить за сущностью некоторую роль, достаточно сопоставить свойства роли с аналогичными свойствами сущности. Таким образом, задача назначения сущности на роль распадается на две подзадачи: задачу выявления и моделирования ролевых характеристик и задачу прикрепления роли к сущности.

В предметных областях в качестве носителей "ролевых" характеристик выступают объекты определенной природы, которые, как и роли, также обязаны своим происхождением сущностям (зависимы от них). Способность сущности осуществлять определенные виды деятельности (обобщенные в виде той же должности) оценивается, главным образом, посредством набора квалификационных параметров. Предметные области включают большое число разнообразных групп объектов, специализирующихся на перечислении таких параметров, значения которых обычно ранжированы по возрастанию или убыванию. "Воинские звания", "ученые степени", "ученые звания", "образовательные квалификации" — примеры подобных типов. Чтобы отличать эти типы от типов других категорий, назовем их "ранговыми" типами (<rank>-типы).

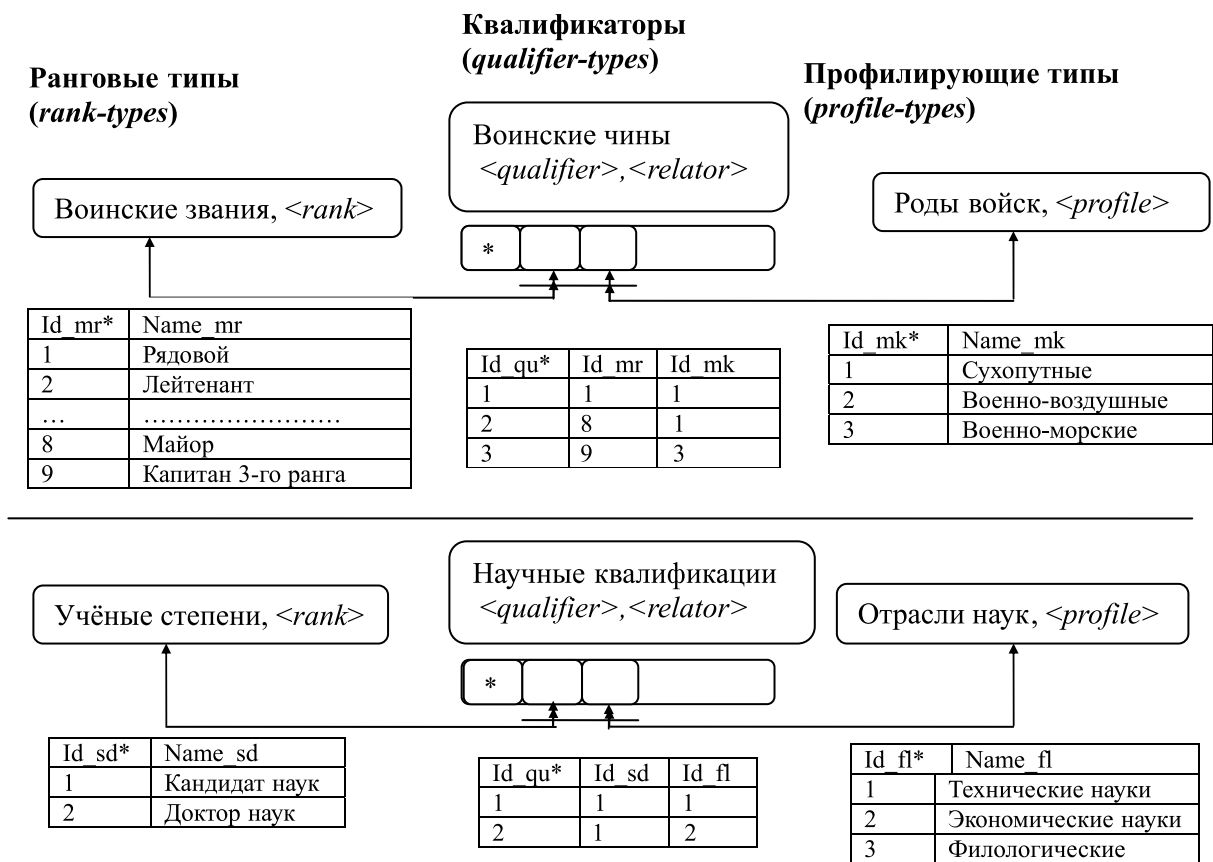


Рис. 8. Квалифицирующий паттерн

Ввиду того что любая профессиональная деятельность осуществляется в специализированных предметных областях, сведения о квалификациях часто дополняются ссылками на сферы профессиональной деятельности, которые, в свою очередь, находят отражение в записях другой категории типов, классифицирующих предметные области или части этих областей. "Роды войск", "Специальности", "Направления подготовки" — примеры таких типов. Поскольку речь в данном случае идет о задании некоего профиля, будем ссылаться на подобные типы как на "профилирующие" типы (<profile>-типы).

Принимая во внимание вышеизложенное, можно прийти к паттерну, моделирующему уровень квалификации сущности (рис. 8).

Как и в случае с *r\_role*-типами, квалификационные характеристики также представляются посредством *relator*. Типы, принадлежащие к этой группе, названы "Квалификаторами". Несмотря на то что пример, приведенный на рис.8, не в полной мере показывает специфику предметных областей (в частности, воинские звания следует дополнительно классифицировать на войсковые и корабельные и установить

связь между ними и родами войск), схема содержит все существенные элементы квалификационного паттерна.

Теперь, используя типы, квалифицирующие роли, можно описать характеристики ролей и ролевые свойства сущностей (рис. 9).

Не вызывает затруднений отразить в модели "квалификационные" характеристики сущностей. Достаточно установить отношения между ними и соответствующими квалификаторами.

Кардинальность отношений между сущностным типом и квалификационными типами в общем случае всегда будет М:М. По крайней мере на это указывают многочисленные факты, взятые из практики. Например, личность может одновременно быть и кандидатом

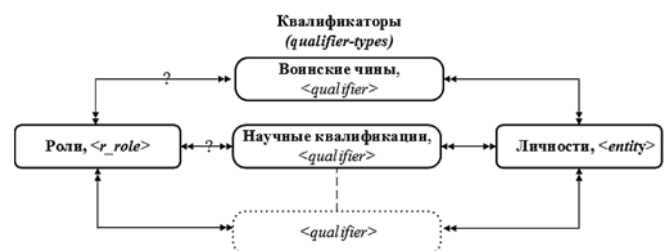


Рис. 9. Соответствия между квалификаторами и сущностями

физико-математических наук, и доктором философии или обладать квалификацией инженера-строителя и магистра экономики.

Отношения же между ролевыми и квалифицирующими типами носят несколько иной характер. Дело в том, что как было указано ранее, роли по своей природе многоальтернативны. Это означает, что роль может задаваться несколькими наборами характеристик. (Каждая характеристика — это пара <характеристика, значение характеристики>.) Кроме того, для конкретизации роли могут привлекаться разные наборы квалифицирующих объектов. Сошлемся для примера на должностные инструкции, содержащие требования к кандидатам на занятие должностей. В них присутствует множество альтернатив, отличающихся как

составом характеристик, принадлежащих разным квалификаторам, так и значениями этих характеристик. Поэтому установления одних только ассоциаций между ролью и квалифицирующими объектами недостаточно. Эти ассоциации не в состоянии аккумулировать все сведения о всех возможных альтернативах. Следовательно, в моделирующем паттерне должен появиться тип — <role\_alt>, содержащий список альтернатив для каждой роли. Представленная ниже подсхема (рис. 10) устраняет указанный пробел.

Теперь для принятия решения о назначении сущности на роль достаточно провести сравнение квалификационных характеристик сущности с характеристиками всех ролевых альтернатив, и в случае совпадения по крайней мере с одной из них закрепить роль за сущностью — установить связь между экземпляром сущности и экземпляром роли.

Обобщим результаты последних двух разделов, схематически представив последовательную эволюцию шаблонов ролевого моделирования (рис. 11).

В практической деятельности можно воспользоваться любым из них, в зависимости от того, на какой уровень функциональности ориентирована информационная система: простейший (рис. 11, а) или полный (рис. 11, в). Последний предполагает охват всей палитры ролевых объектов: ранговых, профилирующих, квалификационных и др.

## Заключение

Принятая в работе трактовка роли как одного из ключевых признаков, на основании которого отбираются объекты, участвующие в определенных классах взаимодействий (другой признак — фазы — в статье упоминается, но не рассматривается), с одной стороны, приводит к единственно возможному способу логического представления роли, а с другой, обнаруживает две взаимодополняющих проблемы:

- поиск способа объединения экземпляров сущностей, принадлежащих разным классам, в один роле-

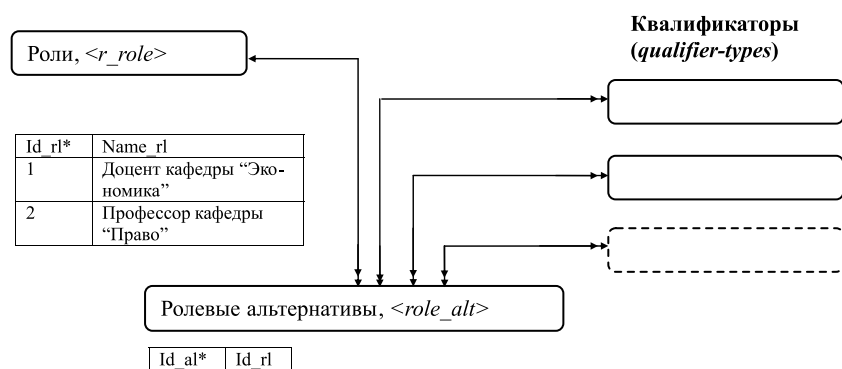


Рис. 10. Шаблон, моделирующий "ролевые" альтернативы

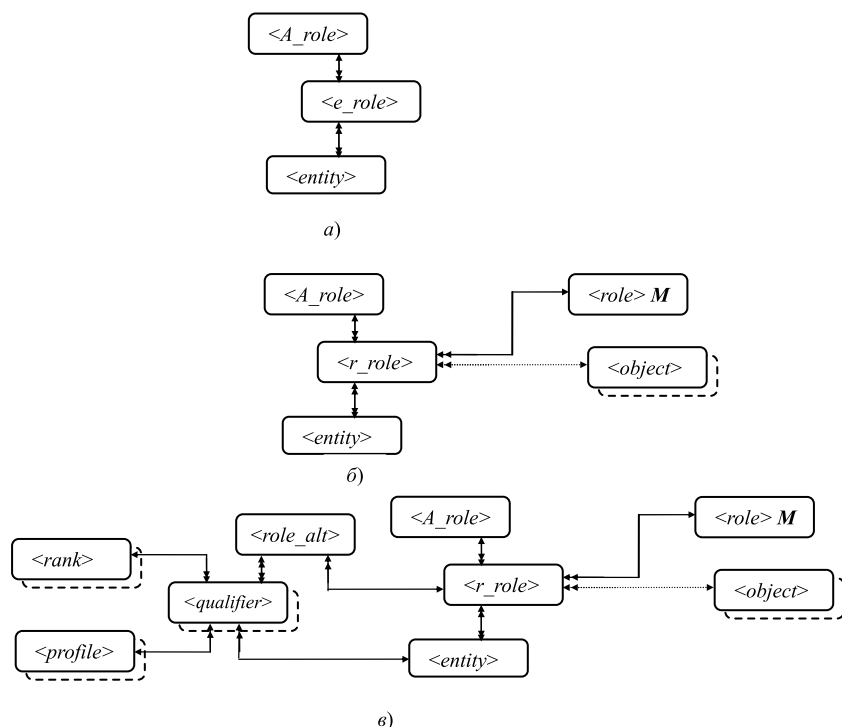


Рис. 11. Эволюция ролевых шаблонов

вой тип, если эти экземпляры в состоянии играть одну и ту же роль. При этом организация логической модели должна исключать появление данных, дублирующих друг друга, и не создавать непреодолимых трудностей доступа к ролевым данным — SQL-запросы не должны содержать явных ссылок на экземпляры ролей;

- неопределенность ролевого прогнозирования, когда требуется отразить в модели экземпляры ролевого типа, которые станут известны только в процессе эксплуатации базы данных.

В работе предлагается и рассматривается способ решения обеих задач, принимающий во внимание два отличительных свойства логической модели: постоянство ее структурной схемы и однозначность идентификации таблиц и ассоциаций, являющихся основными компонентами схемы. Вводится понятие абстрактной роли, и каждой компоненте ассоциации ставится в однозначное соответствие абстрактная роль. Таким образом, список абстрактных ролей становится такой же условной константой логической модели, как и перечень таблиц и ассоциаций. За счет последующей привязки экземпляров реальных ролей к абстрактным достигается и беспрепятственное смешивание сущностей (решается первая проблема), и отпадает необходимость в ролевом предвидении.

Предложены моделирующие подсхемы, организованные в виде шаблонов, что делает их пригодными к практическому применению. Шаблоны включают новые категории типов, вовлеченные в орбиту ролевых взаимодействий: *rank-*, *profile-*, *qualifier-*, *role-alt-* и *r-role-* категории.

Полученные результаты показали тесную взаимосвязь между ролевой и фазной подтипизациями, что может составить предмет дальнейших исследований.

#### Список литературы

1. **Steimann F.** On the representation of roles in object-oriented and conceptual modeling // *Data and knowledge engineering*. 2000. Vol. 35. P. 83–106.
2. **Halpin T., Morgan T.** Information modeling and relational databases. Morgan Kaufmann Publishers, 2008. 970 p.
3. **Буч Г., Рамбо Д., Якобсон И.** Язык UML. Руководство пользователя. М.: ДМК Пресс, 2006. 496 с.
4. **Chen P.** The entity relationship model — toward a unified view of data // *ACM Transaction on Database Systems*. 1976. Vol. 1, N. 1. P. 9–36.
5. **Snoeck M., Dedene G.** Generalization/specialization and role in object oriented conceptual modeling // *Data and knowledge engineering*. 1996. Vol. 19. P. 171–195.
6. **Halpin T. A., Proper H. A.** Subtyping and polymorphism in object-role modeling // *Data and knowledge engineering*. 1995. Vol. 15. P. 251–281.
7. **A. H. M. ter Hofstede and Th. P. van der Weide.** Expressiveness in conceptual data modeling // *Data and knowledge engineering*. 1993. Vol. 10. P. 65–100.
8. **Dahchour M., Pirotte A., Zimanyi E.** A role model and its metaclass implementation // *Information systems*. 2004. Vol. 29. P. 235–270.
9. **Kristensen B. B.** Object-oriented modeling with roles, in: J. Murphy, B. Stone (Eds.), *OOIS'95 // Proceedings of the international conference on object-oriented information systems*, Dublin, Springer. 1996. P. 57–71.
10. **Gottlob G., Schrefl M., Röck B.** Extending Object-Oriented Systems with Roles // *ACM Transactions on Information Systems*. 1996. Vol. 14 (3). P. 269–296.
11. **Elmasri R., Hener A., Weeldreyer J.** The category concept: An extension to the entity-relationship model // *Data and knowledge engineering*. 1985. Vol. 1, N. 1. P. 75–116.
12. **Thalheim B.** Extended entity-relationship model, in: L. Liu, M. T. Ozsu (Eds.), *Encyclopedia of database systems*. Vol. 1. Springer, 2009. P. 1083–1091.
13. **Patig S.** Evolution of entity-relationship modeling // *Data and knowledge engineering*. 2006. Vol. 56. P. 122–138.
14. **Gianni D., Bocciarelli P., D'Ambrogio A.** Referencing capabilities for collaborative engineering of conceptual process modeling with object-role modeling // 2014 IEEE 23-rd international WETICE conference. P. 217–223.
15. **Coulondre S., Libourel T.** Towards a New Role Paradigm for Object-Oriented Modeling // *Proceeding of the OOIS: International conference in object-oriented information systems*. 2002. P. 44–52.
16. **Gogolla M.** UML and OCL in conceptual modeling, in: Embley D. V., Thalheim B. (Eds.) *Handbook of conceptual modeling*, Springer, 2011. P. 85–122.
17. **Keet C. Maria, Fillottrani P. R.** An ontology-driven unifying metamodel of UML Class Diagrams, ERR, and ORM2 // *Data and knowledge engineering*. 2015. Vol. 98. P. 30–53.
18. **Guizzardi G.** Ontological foundations for structural conceptual models. Center for Telematics and Information Technology, University of Twente, The Netherlands, 2005. 441 p.
19. **Carvalho V., Paulo J., Almeida A., Fonseca C. M., Guizzardi G.** Multi-level ontology-based conceptual modeling // *Data and knowledge engineering*. 2017. Vol. 109. P. 3–24.
20. **Родионов А. Н.** Семантическая идентификация, конфигурирование и моделирование типов сущностей в моделях данных // *Вестник НГУ. Серия: Информационные технологии*. 2014. Т. 12, № 1. С. 64–78.
21. **Джексон Г.** Проектирование реляционных баз данных для использования с микроЭВМ. М.: Мир. 1991. 252 с.
22. **Цикритзис Д., Лоховски Ф.** Модели данных. М.: Финансы и статистика, 1985. 168 с.
23. **Cabot J., Raventos R.** Conceptual modeling patterns for roles // *Journal on data semantics V, LNCS 3870*. 2006. P. 158–184.
24. **Parson J., Cole L.** An experimental examination of property precedence in conceptual modeling // *Proceedings of the First Asia-Pacific Conference on Conceptual Modeling (APCCM2004)*. 2004. P. 101–110.
25. **Родионов А. Н.** "Мигрирующие" объекты моделей данных: "слабые сущности" и документы // *Вестник ХГАЭП*. 2011. № 1. С. 40–65.

## The Abstract Roles and the Primitives of Role Modeling in the Conceptual, Logical and Physical Data Models System

*The role concept is regularly used in the development of conceptual data models. With its help, in particular, it becomes possible to dynamically allocate subsets of entities that are allowed to participate in certain interactions. The embodiment of role behavior at the logical level of modeling can be achieved either by multiple cloning of entity's types—one clone per role, or by means of creating specialized role objects and attaching them to certain types of entities. At the same time, both approaches give rise to concomitant problems, which, as in the first case, lead to the uncontrolled growth of clone objects in logical models, or, as in the second, require dynamic assignment to each role of a constant, which will later place explicitly in queries that manipulate database records. The latter will inevitably lead to a violation of the programs and data independence principle. Proposed and implemented in this paper the concept of abstract roles eliminates these shortcomings of role types and leverage on the information contained in logical models. A unified role modeling pattern with a single abstract role type containing a finite list of abstract roles is developed. The list is constant for the selected logical model configuration. The pattern includes real role types, each of which is tied to its own entity type. Such an organization also addresses the issue of forming a virtual set of entities that play the same role, but belong to different entity types.*

**Keywords:** abstract roles, role modeling, role patterns

DOI: 10.17587/it.25.451-466

### References

1. **Steimann F.** On the representation of roles in object-oriented and conceptual modeling, *Data and Knowledge Engineering*, 2000, vol. 35, pp. 83–106.
2. **Halpin T., Morgan T.** Information modeling and relational databases, Morgan Kaufmann Publishers, 2008, 970 p.
3. **Buch G., Rambo D., Jakobson I.** The Unified Modeling Language. User Guide, Moscow, DMK Press, 2006, 496 p. (in Russian)
4. **Chen P.** The entity relationship model — toward a unified view of data, *ACM Transaction on Database Systems*, 1976, vol. 1, no. 1, pp. 9–36.
5. **Snoeck M., Dedene G.** Generalization/specialization and role in object oriented conceptual modeling, *Data and Knowledge Engineering*, 1996, vol. 19, pp. 171–195.
6. **Halpin T. A., Proper H. A.** Subtyping and polymorphism in object-role modeling, *Data and knowledge engineering*, 1995, vol. 15, pp. 251–281.
7. **A. H. M. ter Hofstede and Th. P. van der Weide.** Expressiveness in conceptual data modeling, *Data and Knowledge Engineering*, 1993, vol. 10, pp. 65–100.
8. **Dahchour M., Pirotte A., Zimanyi E.** A role model and its metaclass implementation, *Information Systems*, 2004, vol. 29, pp. 235–270.
9. **Kristensen B. B.** Object-oriented modeling with roles, in: J. Murphy, B. Stone (Eds.), *OOIS'95, Proceedings of the international conference on object-oriented information systems*, Dublin, Springer, 1996, pp. 57–71.
10. **Gottlob G., Schrefl M., Röck B.** Extending Object-Oriented Systems with Roles, *ACM Transactions on Information Systems*, 1996, vol. 14 (3), pp. 269–296.
11. **Elmasri R., Henger A., Weeldreyer J.** The category concept: An extension to the entity- relationship model, *Data and knowledge Engineering*, 1985, v. 1, no. 1, pp. 75–116.
12. **Thalheim B.** Extended entity-relationship model, in: L. Liu, M. T. Oszu (Eds.), *Encyclopedia of database systems*, vol. 1, Springer 2009, pp. 1083–1091.
13. **Patig S.** Evolution of entity-relationship modeling, *Data and knowledge Engineering*, 2006, vol. 56, pp. 122–138.
14. **Gianni D., Bocciarelli P., D'Ambrogio A.** Referencing capabilities for collaborative engineering of conceptual process modeling with object-role modeling, *2014 IEEE 23-rd international WETICE conference*, pp. 217–223.
15. **Coulondre S., Libourel T.** Towards a New Role Paradigm for Object-Oriented Modeling, *Proceeding of the OOIS: International conference in object-oriented information systems*, 2002, pp. 44–52.
16. **Gogolla M.** UML and OCL in conceptual modeling, *Handbook of conceptual modeling*, Springer, 2011, pp. 85–122.
17. **Keet C. Maria, Fillotrani P. R.** An ontology-driven unifying metamodel of UML Class Diagrams, ERR, and ORM2, *Data and knowledge engineering*, 2015, vol. 98, pp. 30–53.
18. **Guizzardi G.** Ontological foundations for structural conceptual models. Center for Telematics and Information Technology, University of Twente, The Netherlands, 2005, 441 p.
19. **Carvalho V., Paulo J., Almeida A., Fonseca C. M., Guizzardi G.** Multi-level ontology-based conceptual modeling, *Data and knowledge Engineering*, 2017, vol. 109, pp. 3–24.
20. **Rodionov A. N.** Semantic Identification, Configuration And Entities Types Modeling For The Data Model Engineering, *Vestnik NGU. Serija: Informacionnye tehnologii*, 2014, vol. 12, no. 1, pp. 64–78 (in Russian).
21. **Dzhekson G.** Relational database design with microcomputer applications, Moscow, Mir, 1991, 252 p. (in Russian).
22. **Cikritzis D., Lohovski F.** Data Models, Moscow, Finansy i statistika, 1985, 168 p. (in Russian).
23. **Cabot J., Raventos R.** Conceptual modeling patterns for roles, *Journal on data semantics V*, LNCS 3870, 2006, pp. 158–184.
24. **Parson J., Cole L.** An experimental examination of property precedence in conceptual modeling, *Proceedings of the First Asia-Pacific Conference on Conceptual Modeling (APCCM2004)*, 2004, pp. 101–110.
25. **Rodionov A. N.** "Migrating" objects of data models: "Weak entities and documents", *Vestnik HGAIeP*, 2011, no. 1, pp. 40–65 (in Russian).