

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

Системная интерпретация ключевых категорий концептуальных моделей баз данных

Исследованы динамические аспекты поведения сущностей в предметной области на основе моделей их собственных жизненных циклов. В роли последних использованы адаптированные процессные модели, содержащие в качестве базовых компонентов события, условия и действия. В контексте переменных свойств, меняющих свои значения в ходе реализации сущностями своих жизненных циклов, рассмотрены, интерпретированы и увязаны между собой информационно-управленческие категории, необходимые для последующего построения структурной компоненты базы данных.

Ключевые слова: жизненный цикл сущности, роли, состояния, действия, метки времени, модель и структуры данных

Введение

Каждая из актуальных задач моделирования данных, будь то задача нарушения хронологической последовательности, адаптивности структур или разбиения множества сущностей на подклассы (типы) или какая-то другая — все они должны опираться на строгую систему взаимоисключающих и в то же время увязанных между собой однозначных понятий, не допускающих двойного толкования. В этом плане применительно к области, связанной с организацией баз данных, просматриваются определенные пробелы, сдерживающие развитие соответствующей теории.

Данные, факты, события, действия, роли, состояния, статусы, условия, метки времени, жизненные циклы, сущности, объекты и свойства — вот список категорий, широко используемых в профессиональном общении и публикациях, затрагивающих область информационных знаний, которую можно обозначить как инженерия данных. За каждой из перечисленных категорий могут скрываться близкие по смыслу, но не идентичные понятия. Например, анализ многочисленных ролевых концепций, сделанный в работе [1], показывает, что роли могут быть интерпретированы как связи между сущностями, использованы в качестве оснований для генерализации/специализации последних и выступать в качестве самостоятельных объектов баз данных.

Аналогичная картина наблюдается и в отношении ряда других категорий моделей и баз данных [2—4]. Более того, практически не ставится и не решается задача по увязке этих категорий между собой.

Постоянно растущие требования к информационным системам (ИС) [5] и в первую очередь к их функциональности, что выражается в постепенном

вытеснении фактографических систем системами управления потоками работ, влекут за собой появление новых информационных объектов и необходимость в их точных определениях.

В настоящей работе введена формальная основа в виде моделей жизненных циклов сущностей (МЖЦС), повторяющих управленческие процессы предметной области (ПО), и на ее базе уточнены и увязаны между собой ключевые понятия даталогического моделирования.

1. База данных как результат отражения взаимодействующих сущностей ПО

Что должна содержать база данных (БД), чтобы считаться полной? В зависимости от того, какой будет ответ на этот вопрос, такой окажется и формальная основа для определения минимального набора взаимосвязанных категорий структурной составляющей БД и интерпретации последних.

Организацию и функционирование произвольной ПО можно представить посредством множества целевых моделей. Структурная компонента БД (модель данных) — одна из таких целевых моделей, в которой находят отражение сущности, образующие ПО, и различные аспекты их взаимодействия между собой. (В дальнейшем будем ссылаться на реляционную модель данных как один из классов логических моделей, обладающий наилучшей формальной основой.)

Собственно самих разновидностей сущностей (типов, множеств сущностей) не так уж и много, как может показаться на первый взгляд. Во всяком случае, их гораздо меньше, чем других категорий объектов, которые также "включаются" в базу данных. Покажем это чуть позже, а пока перечислим

те из типов сущностей, наличие которых можно обнаружить в любой предметной области: личности; товарно-материальные ценности; готовая продукция; финансы. Если говорить о конкретной ПО, то в ней всегда будут присутствовать специфические, характерные только для нее типы. В системах образования это будут дисциплины, виды занятий, учебные планы, в жилищно-коммунальном секторе — оборудование, услуги и т. д.

От сущностей и будем далее отталкиваться. Пока же условимся называть сущности и другие, представляющие интерес и включаемые в базы данных категории, просто объектами. Покажем, что представляет собой ПО, классифицируемая также еще и как эргономическая система, сугубо в "сущностном" аспекте.

У всех систем есть цель. Цель эргономической системы получить какой-то конечный результат в форме готового продукта или услуги, используя (переработав) некоторые ресурсы (исходные сущности). (Экономическая, социальная и прочие составляющие оставлены в стороне, поскольку на первом плане сейчас находятся сущности). Каким образом формируется конечный результат? За счет трансформации (преобразования) исходных или промежуточных сущностей, которые были получены на одной из стадий переработки. (Термин "трансформация" далее будем применять для любых манипуляций с сущностями, в том числе и приводящих к смене их местоположений.)

Продемонстрируем в контексте данных, что происходит с сущностями как в процессе их "переработки", так и по завершении преобразования. Для этого достаточно показать, что собой представляет отдельно взятое переменное свойство сущности (которое в отличие от постоянного может изменить свое текущее значение) и что с ним происходит в результате преобразования.

Концепция свойства, как и необходимость в разбиении множеств объектов на непересекающиеся подмножества для исключения избыточности и обеспечения компактности структур, в которых будут размещены данные, являются ключевыми для понимания и формализации преобразовательных процессов.

Будем придерживаться следующих обозначений для описания соответствующих множеств и свойств элементов, входящих в эти множества. Пусть

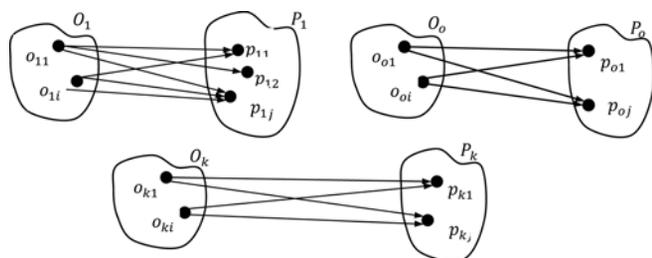


Рис. 1. Соответствия между множествами объектов и категориями свойств

$\Theta = \{O_1, O_2, O_o, \dots, O_K\}$ — совокупность K непересекающихся множеств, образованных объектами ПО. (Каждое такое множество в теории моделирования данных принято называть типом сущности (объекта)). Любому элементу $o_{oi} \in O_o$ ставятся в соответствие все элементы другого множества P_o — множества видов (категорий) свойств, которыми обладают все элементы O_o (рис. 1).

Все переменные свойства, если придерживаться классификации, приводимой в работе [6], могут быть однозначными (принимающими одно единственное значение в произвольный момент времени) или многозначными (имеющими одновременно множество значений). Например, свойство "Наименование" применительно к маркам истребителей в его однозначном и многозначном вариантах будет отличаться количеством значений. СУ-25, Frogfoot и Грач — это альтернативные названия одного и того же советского/российского штурмовика.

Следует различать классы (виды, типы) свойств и их отдельные экземпляры, которые содержат значения, принимаемые свойством в определенные периоды времени. Далее на формальном уровне с использованием небольшого примера продемонстрируем, каким образом задаются класс и экземпляр свойства.

Пусть, например, O_o — это канцелярские принадлежности (КП), o_{oi} — шариковая ручка (ШР), а p_{oj} — свойство, предназначенное для указания местоположения (МП) o_{oi} . Воспользуемся относительным способом задания местоположения, когда сущность позиционируется посредством перечисления других сущностей. В этом случае для задания p_{oj} достаточно просто перечислить элементы множества объектов $\Theta' \subseteq \Theta$, участвующих в формировании свойства, и указать семантику (*significs*) свойства, чтобы исключить потенциальные дубликаты p_{oj} , характеризующиеся тем же самым Θ' . Отсюда, $p_{oj} = (\Theta', \text{significs})$.

Согласно сказанному местоположение КП как класса можно, например, задать так: $p_{КП} \text{ МП} = (\text{Склады}, \text{МОЛ}, \text{Позиционирование})$, где *Позиционирование* — семантика свойства, *МОЛ* — материально-ответственные лица.

Чтобы перейти к значению произвольного p_{oj} потребуется воспользоваться отображением Γ либо множества O_o , либо множества, представляющего декартово произведение O_o с какими-то другими множествами, входящими в совокупность Θ , во множество Q . Последнее в общем случае также является декартовым произведением как минимум трех множеств: одного из множеств значений D_i (домен) и множеств, представляющих метки времени соответственно начала и окончания действия свойств: $T_m^{\text{start}}, T_m^{\text{finish}}$. Индекс m здесь указывает на разновидность используемого множества, содержащего метки времени определенного формата.

Таким образом, $\Gamma_o: O \rightarrow Q$, где $Q = D_i \times T_m^{start} \times T_m^{finish}$, $O = O_o \wedge O_o^s \times \prod_{o \in K} O_o$, где s — степень множества O_o , $s \neq 0$.

Применительно к той же ШР для задания значения, которое ее свойство местоположение может иметь в интервале $[T_m^{start} = 7, T_m^{finish} = 10]$, требуется указать экземпляры объектов множеств *Склады* и *МОЛ*, сославшись на идентификаторы последних.

Поскольку сейчас речь идет не о виде свойства, его значении, то обозначение свойства следует снабдить дополнительным индексом i , ссылающимся на объект. В итоге, значение свойств задаем как: $p_o = \text{кп } i = \text{шр } j = \text{мп} = (\text{Склад} = 10, \text{МОЛ} = 12, 7, 10)$. Далее, в целях удобства индекс o в обозначении свойства будем опускать, и трактовать p_{ij} как j -е свойство i -го объекта.

Для моделирования структур, в которых будут размещаться данные, более удобной формой представления свойства является схема реляционного отношения:

$$R(\text{Свойство})_{ij} = p_{ij} = \{O_i, \bar{O}_i, T_m^{start}, T_m^{finish}, D_i\}, (1)$$

в котором имена атрибутов ассоциируются с перечисленными ранее множествами, а \bar{O}_i — это кортеж произвольной длины, составленный из элементов множества Θ .

Из всего сказанного вытекает, что свойство объекта p_{oj} есть не что иное, как зафиксированный результат взаимодействия этого объекта с другими объектами ПО. Отсюда о траектории объекта в ПО можно судить по тому, в какие моменты времени и на сколько менялись значения его свойств.

Представленная концепция свойства была бы неполной, если бы не был рассмотрен способ фиксации экземпляров свойств в базах данных, который зачастую не отражает истинный характер изменения свойства. На схеме, приведенной на рис. 2, представлены потенциальные траектории свойства и моменты времени, когда происходит запись в БД его новых значений.

Обратим внимание на дискретный характер фиксации, а также моменты времени t_f^1, t_s^2, t_f^3 и t_f^2/t_s^3 , когда свойство мгновенно приобретает новое значение. Относительно t_f^1 и t_f^3 можно однозначно утверждать, что отображение $\Gamma_o: O \rightarrow Q$ перестало существовать. Правда, неизвестно, что предшествовало

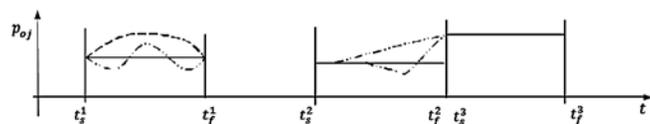


Рис. 2. Варианты динамики свойства сущности

наступлению такого события. Можно также предположить, что некоторые свойства постепенно и вероятно нелинейно меняли свои значения, о чем на рис. 2 говорят штрих-пунктирные и штриховые линии, отражающие характер этих изменений. Но это только предположения, которые если и имели место в ПО, то никакого важного значения для последней не представляли.

Несмотря на то, что соотношение (1) позволяет задать значение любого свойства сущности, требуется выработать и в дальнейшем придерживаться некоторых соглашений, когда нужно отразить факт присутствия или отсутствия сущности в ПО. Свойство "местоположение" как нельзя лучше подходит для этой цели. Достаточно, чтобы все кортежи реляционного отношения, соответствующего схеме (1), для этого свойства имели непустые значения атрибута T_m^{finish} .

Представление в базе данных сущностей предметной области и значений их свойств — это стандартный, минимальный набор требований, на основании его можно восстановить траектории, по которым любая сущность перемещалась в ПО (если траектории представить в виде последовательностей значений свойств). Но очевидно, что интерес представляют не только текущие, но и потенциальные траектории, по которым сущности могут двигаться в будущем, а также силы (механизмы), вследствие которых сущности пришли или придут в движение.

Для решения этой задачи потребуется рассмотреть, исследовать и в какой-то степени дополнить концепцию жизненного цикла сущности.

2. Интерпретация и использование процессных моделей для представления динамики (жизненных циклов) сущностей

Интуитивно (ввиду отсутствия единой точки зрения на модель жизненного цикла сущности — МЖЦС), простейшая МЖЦС должна содержать элементы, отражающие факты пересечения сущностью границ предметной области в прямом и обратном направлении (поступающие в ПО и выходящие из ПО), а также преобразования (переходы) сущности в другие сущности в результате декомпозиции или вхождения в состав таких сущностей. Но подобная модель не охватывает всего многообразия трансформаций, которые происходят или могут произойти с сущностью, и не отвечает на ключевой вопрос о причинах, приводящих к трансформациям.

Поиск сопоставимых по своему целевому назначению моделей из других областей информатики показывает, что примерно аналогичные функции поддерживаются в моделях классов eEPC [7] (событийно-управляемые процессы) и IDEF3 [8] (процессы сценариев), применяемых для описания логики и компонентов бизнес-процессов. События, условия и действия — обязательные атрибуты таких моделей, что дает основание, после незначительных модификаций, рассматривать их в качестве

Образцы ключевых характеристик реальных бизнес-процессов

Трансформируемая сущность	Иницирующие события	Название процесса	Изменяемое свойство
Пассажир Клиент	Запрос на регистрацию Запрос на стрижку	Регистрация пассажира на авиарейс Стрижка клиента в парикмахерском салоне	Регистрация Фасон прически
Покупатель Груз	Выбор приобретаемых товаров Запрос на перевозку груза	Покупка бытовой техники Доставка груза автотранспортной компанией	Приобретенные товары Пункт назначения
Измерительный прибор Упаковочная тара	Запрос на поверку Запрос на изготовление упаковок	Поверка измерительного прибора Изготовление упаковочной тары	Дата очередной поверки Появление новой сущности

МЖЦС. Сделаем выбор в пользу eEPC, как содержащих большее число разновидностей объектов по сравнению с IDEF3 и поэтому позволяющих более точно и детально описать тот или иной бизнес-процесс (БП).

Сформулируем гипотезу, согласно которой любому бизнес-процессу можно поставить в соответствие один и только один тип сущности и трактовать модель этого процесса как последовательность шагов по трансформации одного из свойств, принадлежащих этому типу.

В таблице перечислены БП, в которых специальным образом выделены: тип трансформируемой сущности; свойство, получающее новое значение; запускающее (иницирующее) процесс событие. Таким образом, с формальной точки зрения любой бизнес-процесс задает следующее ключевое преобразование: $p_{ij}(t_s) \xrightarrow{B_r} p_{ij}(t_f)$, где $p_{ij}(t_s)$ — j -е свойство i -й сущности в момент времени начала БП t_s ; $p_{ij}(t_f)$ — то же самое в момент t_f -его завершения; B_r — БП с индексом r .

Обратим внимание на то, что входы всех процессов содержат одно из свойств сущности, принадлежащей некоторому типу, и событие, запускающее процесс, а конечным результатом выполнения является новое значение, приобретаемое свойством.

Если следовать сформулированной гипотезе, то eEPC-модели в первом приближении подходят на роль МЖЦС. Далее в контексте сущностных взаимодействий продемонстрируем, что происходит с сущностями предметной области в процессе перехода от $p_{ij}(t_s)$ к $p_{ij}(t_f)$.

3. Компоненты жизненных циклов сущностей: состояния, события, действия, условия

Совокупность значений, принимаемых свойствами сущности в произвольный момент времени, определяет то, что принято считать состоянием [2]. Таким образом, состояние i -й сущности в момент времени t — $S_i(t)$ формально задается кортежем $S_i(t) = (p_{i1}(t), p_{i2}(t), \dots, p_{ij}(t))$, где $p_{ij}(t)$ — значение j -го свойства в момент t . О целесообразности отслеживания $S_i(t)$ будет сказано ниже, а пока отметим, что одной единственной характеристики, даже такой обобщенной, каковой является состояние, явно

недостаточно для описания поведения объектов, которое по своей природе динамично и непрерывно. Важно также знать, каким образом менялось то или иное свойство. Эту функцию в ИС выполнял действия. Действия, таким образом, в какой-то степени подменяют собой непрерывность. Покажем в общем виде, что происходит с сущностью в промежутке времени, когда одно из ее свойств меняет свое значение. Наглядное представление об этом дает срез фрагмента бизнес-процесса (рис. 3), на котором представлены только объекты, участвующие в серии взаимодействий, повлекших за собой изменение их свойств.

Пусть значение j -го свойства сущности с индексом i в промежутке времени $[t_s, t_f]$ изменилось на величину $\Delta p_{ij} = p_{ij}(t_s) - p_{ij}(t_f)$. Поскольку, как ранее было замечено, любое свойство — это результат взаимодействия сущности с другими сущностями (в общем случае с другими объектами), то Δp_{ij} могло произойти только, если в промежутке $[t_s, t_f]$ менялись какие-то другие свойства $j' \neq j$ i -й сущности и (или) свойства других сущностей $i' \in i, i' \in I, j, j' \in J$, где I, J — множества индексов сущностей и свойств i -й сущности соответственно. Таким образом, всякому изменению свойства p_{ij} , которое (изменение) для удобства обозначим как \vec{p}_{ij} , может быть поставлена в соответствие последовательность $\{\vec{p}_{ij}^s\}$, где s — порядковый номер элемента последовательности, образованной модифицируемыми свойствами как i -го объекта, так и свойствами других i' -х объектов. Более того, таких последовательностей, участвующих в достижении новых значений p_{ij} , может быть несколько. Например, процессы регистрации пассажиров в разных аэропортах на рейсы разных авиакомпаний могут различаться друг от друга, что, соответственно, порождает и уникаль-



Рис. 3. Последовательность модифицируемых свойств, "участвующих" в достижении $|\Delta p_{ij}|$

ные $\{\vec{p}_{ij}\}_k$. (Здесь k обозначает индекс подобной последовательности.)

Структура (содержание) конкретной последовательности зависит только от самого факта изменения значения свойства. Поэтому не имеет принципиального значения приращение свойства $[\Delta p_{ij}]$.

Среди элементов, образующих $\{\vec{p}_{ij}^s\}$ последовательности, встречаются не только свойства, но и встроенные $\{\vec{p}_{i'j}^s\}$ или $\{\vec{p}_{ij'}^s\}$. Данное обстоятельство отражает факт того, что в рамках основного процесса разворачиваются отдельные, самостоятельные подпроцессы.

Сформулируем и запишем очевидные темпоральные ограничения, которые должны выполняться для $\forall\{\vec{p}_{ij}^s\}$.

1. Обозначим через $[t_s, t_f]_{p_{ij}} = Y_{ij}$ интервал изменения p_{ij} . Первое ограничение состоит в том, что

$\forall [t_s, t_f]_{p_{ij}}^{n_{ij}} \in Y_{ij} \wedge \forall [t_s, t_f]_{p_{i'j}}^{n_{i'j}} \in Y_{ij}$, где $n_{i'j}$ и $n_{ij'}$ — порядковые номера интервалов изменения соответственно p_{ij} и $p_{i'j}$, $n_{i'j} = 1 \dots N_{i'j}$; $n_{ij'} = 1 \dots N_{ij'}$. $N_{i'j}$, $N_{ij'}$ — общее число таких интервалов. Другими словами, вес промежутки времени, в течение которых происходит изменение любого из свойства последовательности $\{\vec{p}_{ij}^s\}$, должны лежать в границах Y_{ij} .

2. Если какое-то из свойств $\{\vec{p}_{ij}^s\}$ несколько раз меняет свое значение в интервале Y_{ij} , т. е. $n \neq 1$, как,

например, p_{ij} на рис. 3, то все интервалы $Y_{ij}^{n_{ij}}$ не

должны пересекаться, $\bigcap_{n_{ij}=1}^{N_{ij}} Y_{ij}^{n_{ij}} = \emptyset$. Сказанное свидетельствует о том, что свойство не может быть подвергнуто многократному изменению в одном и том же временном диапазоне. Например, "Преподаватель не может одновременно читать две и более лекции, один и тот же самолет не в состоянии обслуживать рейсы, которые выполняются в одно и то же время."

3. Третье ограничение касается величины "технологического" разрыва Δ_{ij} (см. рис. 3), который может иметь место для отдельных свойств сущности. Если, например, занятия со студентами проводят в аудиториях, расположенных в разных корпусах, то необходимо время для перемещения студентов из одного корпуса в другой корпус. (Более детальное рассмотрение этого ограничения ввиду его очевидной значимости и сложности моделирования составляет предмет отдельного исследования.) Заметим, что последнее ограничение требует фиксации Δ_{ij} в метаструктурах базы данных.

Перечисление одних только последовательностей $\{\vec{p}_{ij}^s\}$, несмотря на то, что они показывают траекторий, по которым меняется то или иное свойство сущности, не дает ответа на вопрос о том, что происходит с сущностью в промежутке $[t_s, t_f]$. Как было отмечено ранее, эта роль возлагается на дей-

ствия. Это могут быть как действия, выполняемые самой i -й сущностью, так и действия, совершаемые другими i' -ми сущностями, воздействующими на i . Таким образом, каждое действие $a_{ii'}^h$ кроме очевидного названия, характеризуется также источником и адресатом. Здесь h — индекс действия; i — ссылка на объект, выступающий в роли источника; i' — индекс объекта, на которое направлено действие.

В отличие от показанного ранее соответствия (см. рис. 1) между множествами объектов и свойств в отношении действий уместно говорить о композиции двух отображений: $D_1: \Theta \rightarrow A$ и $D_2: A \rightarrow \Theta$, где A — множество действий. В результате получаем $D: \Theta \rightarrow \Theta$.

Подчеркнем, что действия могут совершать не все объекты, а только их часть в лице сущностей (что в общем-то очевидно, так как другие категории объектов являются в некотором роде производными от сущностей). В свою очередь, адресатами действий могут выступать все объекты, а не только сущности.

Используя концепцию действий, рассмотрим детально элементарное \vec{p}_{ij} (не включающее, как было отмечено ранее, собственные $\{\vec{p}_{ij}\}_k$). Вообще говоря, интерес представляют только несколько моментов, которые сформулируем в виде вопросов.

1. Вследствие того, что свойство некоторого объекта p_{ij} согласно (1), предполагает указание как исходного объекта, так и перечисление объектов, участвующих во взаимодействии с исходным, могут ли в изменении этого свойства принимать участие третьи объекты?

2. Может ли \vec{p}_{ij} произойти только в результате взаимодействий исключительно третьих объектов (без участия исходного и вторичных объектов)?

Ответы на оба вопроса будут отрицательными. Если предположить, что отыщутся некие третьи объекты, влияющие посредством своих действий на \vec{p}_{ij} , то они также будут вынуждены изменить какие-то свои свойства. Но в этом случае \vec{p}_{ij} не будет элементарным.

Представленные рассуждения позволяют определиться (рис. 4) с частными (а, б, в) и общим (г)

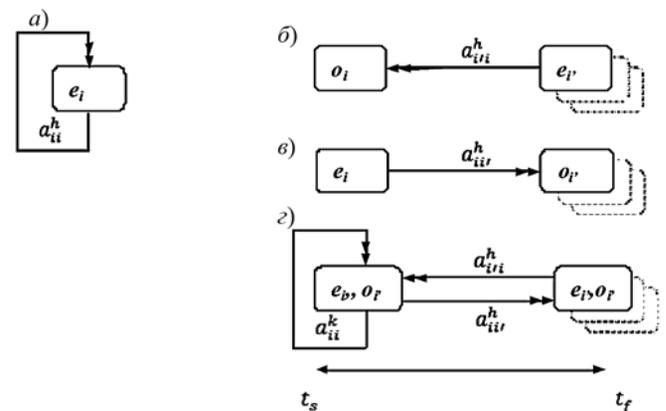


Рис. 4. Классы взаимодействий между объектами

классами взаимодействий между сущностями посредством действий, приводящих к модификации значений их свойств. Сущности здесь помечены символом e . (Далее везде для обозначения экземпляров и типов будем использовать соответственно строчные и прописные буквы.)

Увязав между собой объекты, их свойства и действия, как элементы динамики свойств, можно, опираясь на упомянутые выше eEPC-модели, приступить к последовательному поэтапному построению МЖЦС. Ограничимся на первом этапе только действиями, сделав акцент на их альтернативность и множественность в реализации \vec{p}_{ij} .

Фрагмент процесса (рис. 5) иллюстрирует две альтернативные последовательности действий для получения фотографического снимка. Для каждого действия можно указать объект-источник и объекты, на которые направлены действия. Выполнение любой их двух последовательностей $\{a^1, a^2, a^3, a^4\}$ или $\{a^1, a^2, a^3, a^5\}$, приводит к получению цифровой фотографии, что в контексте данных трактуется как изменение одного из свойств фотографии, ранее обозначенного как "местоположение".

Этот небольшой фрагмент достаточно информативен, так как дает дополнительную информацию о действиях. Помимо исходных и целевых объектов, в действиях a^4 и a^5 явно присутствуют объекты, посредством которых выполняются действия — фотоаппарат и смартфон соответственно.

Таким образом, каждое действие характеризуется своей семантикой, сущностью, которое его осуществляет, подмножеством сущностей \hat{E} , посредством которых оно выполняется, подмножеством объектов \hat{O} , на которые направлено, а также моментами времени начала и окончания действия:

$$\text{Действие} = (\text{significs}, e_i, \hat{E}, \hat{O}, T_m^{\text{start}}, T_m^{\text{finish}}). \quad (2)$$

Если посредством действий можно показать, что делает или не делает сущность в предметной области в произвольный момент времени, то закономерно возникает вопрос, касающийся того, что приводит сущность в "движение" или по-другому, что или кто активизирует каждое действие, результатом которого является новое значение свойства

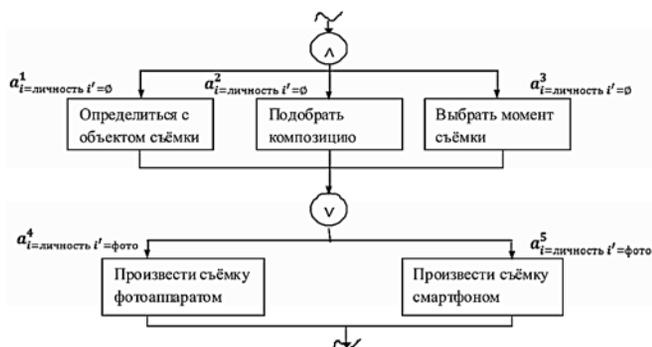


Рис. 5. Альтернативные последовательности действий

сущности. Теория моделирования бизнес-процессов [7, 9] говорит о том, что инициатором любого действия является событие или совокупность событий.

В научной литературе можно встретить множество определений события. Применительно к моделированию данных событие обычно трактуется как мгновенный факт [4]. Факты, в свою очередь, могут ассоциироваться с \vec{p}_{ij} , как это предлагается в работе [2]. Например, к мгновенным фактам (событиям) в [2] относят продажи, покупки, платежи.

В то же время события, если считать их элементами управления (а не только мгновенными фактами, что является только формой их проявления), не сводятся исключительно к изменению свойства \vec{p}_{ij} . Раскроем механизм функционирования событий, для чего исследуем отношения, возникающие между событиями и действиями, событиями и сущностями, действиями и сущностями. Как и в случае с действиями, будем также опираться на организацию моделей бизнес-процессов, принятых ранее в качестве прообразов моделей жизненных циклов сущностей.

В таких моделях могут быть выделены несколько групп событий, которые инициируют действия. Это события V^E , источники которых сущности. Например, "Клиент выразил желание заключить договор аренды помещения", "Воздушное судно запросило разрешение на посадку", "Банк получил платежное поручение, на основании которого он должен провести транзакцию". Во всех случаях явно присутствуют один или несколько сущностных адресатов (агентов).

Другую группу составляют "временные" события V^T . Например, наступило время "Ч" и активизировалось событие, запустившее процесс сдачи экзамена.

В отдельную группу V^S сводятся события, возникающие, когда сущности достигают каких-то определенных, характерных состояний. "Клиент открывает вклад на сумму 700 000 рублей", "Величина прожиточного минимума составила 7253 рубля", "Объем заказа превысил установленный максимум", "Уровень текущих запасов достиг критических (страховых) значений". (Обратим внимание на то, что здесь фигурируют состояния, вопрос о целесообразности использования которых поднимался ранее.)

Еще одну группу составляют события, сигнализирующие о том, успешно завершилось то или иное действие или нет, $V^R = \{true, false\}$.

Чтобы каким-то образом различать результирующие события, которые "привязаны" непосредственно к действиям и выступают для них одновременно в роли входных и выходных событий, и события, составляющие V^E , V^T и V^S , являющиеся по отношению к действиям всегда входными, будем последние называть иницилирующими событиями V^I , $V^I = V^E \cup V^T \cup V^S$.

На схеме (рис. 6) показаны стандартные отношения, возникающие между действиями и событиями.

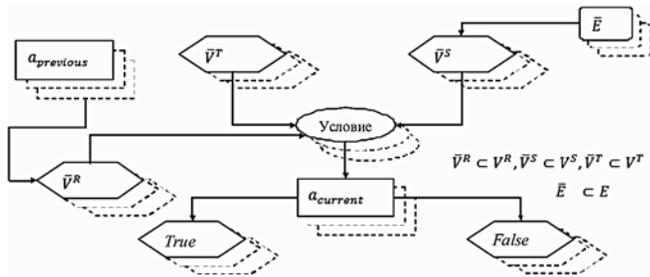


Рис. 6. Отношения между событиями и действиями



Рис. 7. Входы и выходы условия как объекта МЖЦС

В соответствии с представленными конфигурациями каждому действию предшествует одно или несколько инициирующих событий и (или) события, сгенерированные уже завершившимися действиями.

Показанные соответствия между множествами событий, действий и сущностей очевидны и вытекают из природы поведения сущностей в организационно-экономических системах, в которых ничего кроме сущностей и взаимодействий последних друг с другом обнаружить нельзя.

Осталось определиться с условиями, которые задают альтернативные траектории "движения" сущностей в рамках их жизненных циклов, а также с подмножеством "пустых" действий, которые не оказывают никакого воздействия на свойства сущностей.

"Имеется ли бронь на гостиничный номер?", "Есть ли на складе достаточный запас товара?", "Какова фактическая стоимость кредита?". Это все примеры "пустых" действий. Их объединяет то, что они сводятся к получению либо информации непосредственно о текущих свойствах сущностей, либо информации, которая может быть выведена на основании тех значений, которые принимали свойства. (Схема реляционного отношения, которая будет представлена ниже, позволяет фиксировать такие действия, не прибегая к использованию дополнительных отношений.)

Реальные сущности в границах моделей своих жизненных циклов могут "двигаться" по различным, альтернативным траекториям. "Развилочную" миссию выполняют условия, функционирование которых можно условно представить в виде "черного" ящика (рис. 7). Входы для условий — это разнообразные события, выходы — инициируемые действия.

С точки зрения математической логики любое событие представляет собой высказывание, которое всегда истинно. Тогда условие, в свою очередь, — это сложное высказывание, полученное из простых высказываний посредством логических связок "не", "и", "или". (Заметим, что импликация и эквиваленция, которые также используются для построения сложных высказываний в eEPC не используются.)

4. Факты и привязка ко времени объектов, событий и действий

Моделирование событий, действий и условий было бы неполным, если бы не был затронут хотя бы в общих чертах фактор времени.

Формально метка времени — это одна или несколько единиц измерения времени (квантов времени C), поставленных в соответствие какому-то информационному объекту: свойству сущности, событию или действию. Речь идет о кортежах вида: $(o_i, c_1, c_2, \dots, c_h)$, $(v_i, c_1, c_2, \dots, c_h)$, $(a_i, c_1, c_2, \dots, c_h)$. Собственно говоря, факты — это и есть подобные кортежи.

Обратим внимание на то, что условия не требуют привязки ко времени. Будем полагать, что, во-первых, они срабатывают мгновенно, а во-вторых, их окружают события и действия, которые уже снабжены метками времени и поэтому несложно установить момент времени срабатывания каждого конкретного условия.

Очевидно, что если подходить к вопросу привязки системно, то за интегрирующую основу логично принять МЖЦС, объединяющую все перечисленные в заголовке раздела категории.

Какой бы аспект моделирования данных не затрагивался, всегда необходимо проводить различие между моделированием типов и моделированием их экземпляров, а также устанавливая отношения между получаемыми моделями. Будем и здесь придерживаться этой схемы. Отметим наиболее значимые моменты. Как и в отношении объектов, действий и событий МЖЦС также разбиваются на пересекающиеся множества, именуемые типами, а их элементы считаются экземплярами. Например, для типа процесса "Проведение медицинского освидетельствования участников дорожного движения" всегда будет иметь место множество реализаций, число которых будет увеличиваться по мере появления очередного "клиента". Далее, под МЖЦС всегда будет подразумеваться тип, а ЖЦС ассоциироваться с экземпляром.

По понятным причинам в составе МЖЦС в качестве их элементов могут присутствовать только типы (объектов, действий и условий). Соответственно ЖЦС будут всегда содержать только экземпляры объектов, действий и условий. Вероятно, что должны существовать и различия в форматах и механизмах представления и привязки ко времени объектов, входящих в состав МЖЦС и ЖЦС. В МЖЦС всегда будет фигурировать абстрактное относительное (нереальное) время ("начало учебного года", каждая вторая неделя месяца и т. п.). Тогда как в процессе отслеживания реальных траекторий сущностей (в ЖЦС) будет задействовано фактическое, астрономическое время. Наглядное представление об использовании меток времени в МЖЦС дает схема, приведенная на рис. 8. На ней присутствуют две альтернативные метки времени D_f и C_h , а также N_c , являющаяся производной от

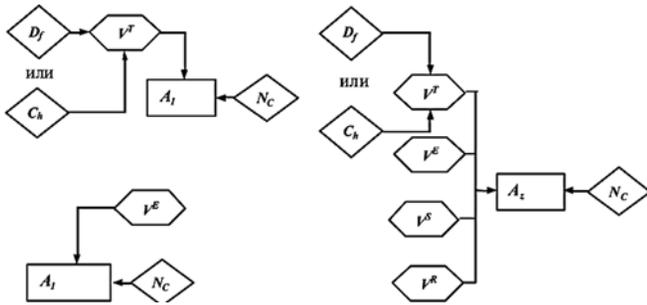


Рис. 8. Допустимые метки времени для стандартных конфигураций событий и действий в МЖЦС

них. Раскроем содержание и назначение всех перечисленных форматов.

Будем руководствоваться тем, что любая единица измерения времени всегда характеризуется продолжительностью, а совокупности таких единиц образуют последовательности (месяцы года, дни недели, недели года и т. д.). Поэтому для моделирования относительных времен достаточно сослаться на порядковый номер кванта h в такой последовательности C_h . Но можно, если база данных "поддерживает" требуемую для моделируемого процесса разновидность квантов — "временной" тип (формат) данных D , воспользоваться одним из таких форматов D_f , где f — порядковый номер формата. Предположим, что "встроенный" формат имеет размерности $hhmm$ (часы-минуты). Тогда $D_f = hhmm = 12.00$ будет означать, что событие произойдет в 12 часов дня.

Иницирующие события в зависимости от их класса (сущностей V^E , времени V^T и состояний V^S) на первый взгляд могут соотноситься как с квантами C_h , так и с временными типами данных D_f . Но для элементов V^E и V^S , ввиду случайного характера их наступления, метку времени ни в виде D_f , ни посредством C_h , задать нельзя. Что касается V^R , то оно наступает в момент завершения действия и может быть получено расчетным путем.

Вне зависимости от того, используется ли в качестве меток времени для событий D_f или C_h , продолжительность любого действия N_c всегда будет измеряться числом квантов, в течение которых это действие будет выполняться. Альтернативный для N_c вариант с указанием времени начала действия отпадает ввиду того, что формально МЖЦС — это граф, а в графе присутствует множество путей разной длины (продолжительности) достижения конкретного действия. Поэтому нельзя точно определить, когда начнется текущее действие A_z .

В отношении исходных действий A_1 также не имеет смысла задание времени их начала, поскольку оно совпадает с меткой времени события V^T .

Измерение N_c посредством квантов в случае использования D_f для событий, актуализирующих действия, приводит к необходимости решения задачи определения размера кванта для выбранного D_f . Перечислим варианты, которые могут возникнуть. Иницирующее D_f задействует одну единицу изме-

рения, а продолжительность действия — другую. Например, D_f — это часы, а действия — минуты. Поскольку подобных несоответствий в МЖЦС скорее всего окажется несколько, приемлемым решением был бы выбор наименьшего используемого в МЖЦС кванта и приведение всех D_f и C_h к этому значению. (В противном случае невозможно определить время завершения каждого действия и всего процесса в целом.) Если придерживаться этого соглашения, то вид кванта h становится одной из важнейших характеристик процесса, определяя по сути все возникающие в последнем темпоральные ограничения.

Появление в МЖЦС V^T -событий, которые активизируют промежуточные и завершающие действия A_z , при некорректном задании D_f или C_h может привести к старту текущего A_z при еще не завершившихся предшествующих действиях. Для примера возьмем процесс "Проведение научной конференции", в котором пленарные и секционные заседания начинаются в фиксированные времена и первый доклад в секционной части может начаться не раньше завершения последнего доклада в пленарной части.

Поскольку на графе МЖЦС к произвольному A_z может вести несколько путей, то длину максимального пути можно использовать для формализации ограничения, исключающего возникновение подобных ситуаций.

Пусть P_z — множество путей, ведущих к A_z , каждый из которых характеризуется длиной, измеряемой числом квантов. $L = \max P_z$ — наибольший элемент множества P_z . Тогда $C_h(V_z^T(A_z)) \geq L$.

В отличие от МЖЦС отслеживание траекторий реальных сущностей, которые протекают в границах этих МЖЦС, ведется уже в реальном масштабе времени, что исключает использование C_h . Все современные СУБД поддерживают встроенный "временной" формат вида $uuuymmddhhss$ и содержат семейство "временных" функций, позволяющих получить из указанного формата широкий диапазон разновидностей квантов. Тем самым, практически исключается необходимость в поддержке каких-то уникальных квантовых форматов.

Соответствия, возникающие между метками времени d_f и событиями, d_f и действиями в ЖЦС, приведены на рис. 9.

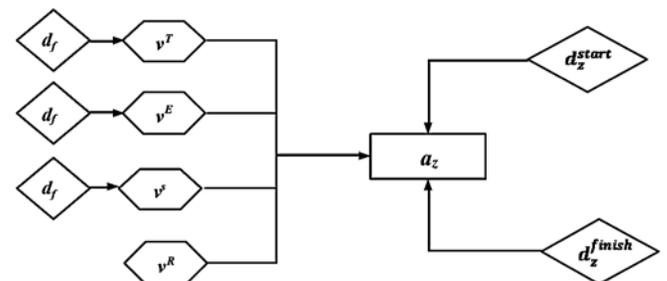


Рис. 9. Допустимые метки времени для событий и действий в ЖЦС

Представленные здесь отображения ($a_z \rightarrow d_z^{start}$ и $a_z \rightarrow d_z^{finish}$) справедливы для $\forall a_z; d_z^{start}$ и d_z^{finish} задают моменты времени соответственно начала и завершения действия a_z .

5. Условия в качестве объектов базы данных и структуры, моделирующие МЖЦС

Вопрос, на который предстоит получить ответ, формулируется следующим образом: нужно ли хранить в базах данных условия и если да, то каким образом это следует делать?

Для отслеживания траекторий реальных сущностей нет необходимости в фиксации условий, потому что интерес представляют только те факты, в какой последовательности, когда и на какое значение менялись свойства сущностей. Этот минимальный набор требований, который выдвигают учетные ИС, именуемые еще и как системы обработки транзакций. Повышение ИС в классе, например, переход их в разряд систем принятия решений, влечет за собой расширение исходного набора требований, который может быть дополнен, в том числе, и требованием отслеживания потенциальных траекторий трансформации сущности. Такая информация может быть получена, если только в базе данных хранится непосредственно сама модель жизненного цикла, элементами которой в том числе являются и условия.

Поскольку условие в общем виде задается с помощью сложного высказывания (например, "Размер вклада лежит в пределах от 100 до 700 тысяч рублей"), оно может быть определено на домене символов (символьной строки).

Далее покажем, к какой структурной организации (модели данных) приводят результаты, полученные в ходе рассмотрения динамики сущностей в ПО. Ограничимся простейшей подсхемой (рис. 10), отражающей только МЖЦС в ее базовой конфигура-

ции. Имена атрибутов с надстрочным индексом "*" обозначают первичные ключи. Атрибуты, не содержащие индексов, соответствуют внешним ключам.

Часть структур и большинство атрибутов не вошли в модель. Например, на схеме отсутствуют отношения, в которых перечисляются роли, несмотря на то, что в определенных случаях они подменяют классы сущностей в МЖЦС. Не отражены ретроспективные и перспективные модели жизненных циклов и некоторые другие не менее значимые моменты.

Тем не менее, на схеме представлены структуры, образующие ядро МЖЦС-кластера. Посредством этих структур в базе данных может быть размещена информация об узлах и связях графа, каковым с формальной точки зрения является любая МЖЦС.

Структурный кластер, представляющий, в отличие от МЖЦС, экземпляры последних, будет иметь еще более сложную организацию, в том числе включать типы сущностей, характерные для конкретной ПО.

6. Локализация множества сущностей. Исключение ролей и услуг

В начале работы говорилось о том, что разновидностей того, что принято называть сущностями, которые часто ассоциируются с именованными структурами баз данных, в любой ПО на самом деле не так уж и много. За сущности могут приниматься объекты, выполняющие классификационные функции, "слабые сущности", которые на самом деле представляют собой свойства, документы, большинство из них содержат данные о взаимодействиях сущностей (часть документов "заменяют" композитные, составные сущности), и ряд других объектов. Все они, без сомнения, важны и необходимы, но в предметных областях демонстрируют отличное от сущностей поведение, выполняя, по сути, вспомогательную функцию.

К сущностям часто причисляют и услуги, хотя природа последних как самостоятельной категории понятна не до конца. Включение в модель данных структур, представляющих МЖЦС и их экземпляры, дает основание представить новое толкование услуги, которое, с нашей точки зрения, наиболее точно отражает ее суть.

Услуга — это именованная МЖЦС в форме бизнес-процесса. Действительно, если обратиться к приведенной выше таблице, то название каждого из процессов не что иное, как название услуги. (По крайней мере, в той общепринятой манере, как их (услуги) принято обозначать). Таким образом, услуги — это не

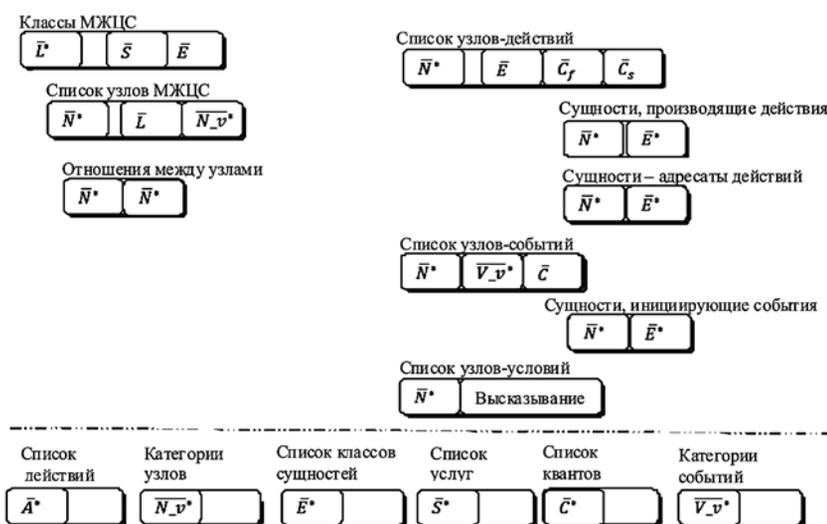


Рис. 10. Структурная организация МЖЦС (концептуальная модель)

сущности. Тем не менее, в реальных базах данных они часто включаются в состав отношений, где перечисляются сущности и задаются характеристики последних. Это своего рода компромисс, типичный для ИС, когда в границах одной и той же ИС одновременно сочетаются и классические учетные фрагменты, и компоненты систем более высокого уровня. Очевидно, что в ИС, поддерживающих исключительно бизнес-процессы, потребность в фиксации услуг отпадает сама собой.

Еще одно подмножество объектов, которые не принадлежат к сущностям, — это роли. Потребность в ролевой концепции возникает всякий раз, когда трансформация сущности в предметной области не ограничивается ее одним единственным жизненным циклом. Возьмем для примера некоего абстрактного человека, являющегося элементом типа сущности "Личности" и работающего в университете. Он может быть преподавателем и в то же время получать *n*-е высшее образование в том же университете, рецензировать научную работу, участвовать в работе диссертационного совета и т. д. Одним словом, "играть" несколько ролей в соответствии с алгоритмами, которые задаются определенными жизненными циклами.

Таким образом, в представленном контексте роли и должности — это тождественные понятия. И такого рода объекты можно обнаружить во многих предметных областях.

Если опираться на только что представленное назначение ролей, то не составляет труда установить два важных для полноты отражения предметной области соответствия: между элементами множеств бизнес-процессов и экземпляров ролей с одной стороны, и между множествами ролей и подмножеством экземпляров реальных сущностей, с другой стороны (рис. 11).

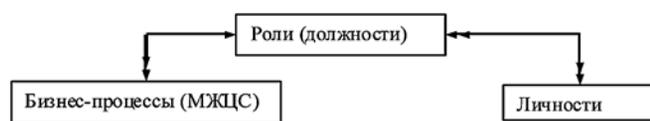


Рис. 11. Стандартные отношения между ролями, типами сущностей и МЖЦС

Тем самым удастся показать, что та или иная сущность (личность в рассматриваемом случае) может параллельно участвовать в нескольких процессах.

Использование ролей в качестве посредника между МЖЦС и экземплярами сущностей обладает еще одним положительным эффектом. Не составляет труда получить ответ на такой вопрос: может ли сущность одновременно участвовать в нескольких процессах? Постановка и решение данной задачи

выходит за рамки статьи и может составить предмет отдельного исследования.

Заключение

Установление взаимно однозначного соответствия между бизнес-процессами и жизненными циклами сущностей формирует основу для более точной интерпретации и взаимной увязки ключевых понятий, используемых в моделях данных и методах их проектирования. Исследование кинематических и динамических аспектов поведения сущностей в ПО через призму процессов раскрывает "движущие силы" в виде событий, условий и действий, приводящие к разноплановым трансформациям сущностей. Результатом таких трансформаций становятся новые значения, которые приобретают переменные свойства сущностей. При этом переход к новым значениям происходит вследствие серий взаимодействий, в том числе и среди сущностей, которые напрямую, через бизнес-процесс, не вовлечены в трансформацию.

Учет в моделях данных динамики сущностей позволяет также однозначно определиться с содержанием таких категорий, как услуги и роли. Услуги ассоциируются с названиями МЖЦС, а роли, которые привязываются к сущностям, однозначно задают подмножество МЖЦС.

Полученные результаты делают возможным решение, по крайней мере, двух актуальных задач моделирования данных: разработку концептуальных схем, исключающих нарушение темпорального порядка фактов в процессе регистрации последних в структурах базы данных и моделирования минимальных временных технологических заделов, определяющих готовность сущности участвовать в новых взаимодействиях.

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. **Jimenez L. J.** REEEM: Reenhancing the entity-relationship model // *Data and knowledge engineering*. 2006. Vol. 58. P. 410–435.
3. **Date C. J., Hugh Darwen, Nikos A.** Lorentzos Time and relation theory. — Morgan Kaufmann Publishers, 2014. 560 p.
4. **Gregersen H., Jensen C. S.** et al. The consensus glossary of temporal database concepts. *Temporal Databases: Research and Practice*. Springer, 1998. P. 367–405.
5. **McGinnes S., Kapros E.** Conceptual independence: A design principle for the construction of adaptive information systems // *Information system*. 2015. Vol. 47. P. 33–505.
6. **Родионов А. Н.** Качество дагалогических схем. Принцип компактности моделей данных и его приложения II // *Информатика и системы управления*. 2012. № 2. С. 14–25.
7. **Шеер А.** Моделирование бизнес-процессов. М.: Весть-МетаТехнология, 2000. 207 с.
8. **Елиферов В. Г., Репин В. В.** Бизнес-процессы: Регламентация и управление. М.: ИНФРА-М, 2011. 319 с.
9. **Вил ван дер Аалст, Кейс ванн Хей.** Управление потоками работ: модели, методы и системы. М.: ФИЗМАТЛИТ, 2007. 316 с.

System Interpretation of Conceptual Data Model Key Categories

This paper examines dynamic aspects of entities behavior at domain of interest by means of original entities lifespan models (ELM). After some supplements to extended event driven process chain (eEPC) model the last uses as a basic for ELM. Such ELM enables to determine entities trajectories and objects which compel entities to migrate with the confine of area. Dynamic of entity interprets as property changes due to interaction of an entity with another objects including entities. Any ELM associates with one and only one property which have to be modified. Actions, events and conditions as components of an entity lifespan are connected with objects and timestamps. Thereby all data model categories become interlocking and pursue the aim to derive more detail and exact database. Addition of unified structures for recording ELM and its instances in data model permits to construe some information objects as roles.

Keywords: entity lifespan, roles, states, actions, timestamps, data model, data structures

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. **Jimenez L. J.** REEEM: Reenhancing the entity-relationship model, *Data and knowledge engineering*, 2006, vol. 58, pp. 410–435.
3. **Date C. J., Hugh Darwen, Nikos A.** *Lorentzos Time and relation theory*, Morgan Kaufmann Publishers, 2014, 560 p.
4. **Gregersen H., Jensen C. S.** et al. The *consensus glossery of temporal database concepts*. Temporal Databases: Research and Practice, Springer, 1998, pp. 367–405.
5. **McGinnes S., Kapros E.** Conceptual independence: A design principle for the construction of adaptive information systems, *Information system*, 2015, no. 47, pp. 33–505.
6. **Rodionov A. N.** Kachestvo datalogicheskikh shem. Princip kompaktnosti modelej dannyh i ego prilozheniya II, *Informatika i sistemy upravleniya*, 2012, no. 2, pp. 14–25.
7. **Sheer A.** Modelirovanie biznes-processov, Moscow, Vest'-MetaTehnologija, 2000, 207 p.
8. **Eliferov V. G., Repin V. V.** *Biznes-processy: Reglamentacija i upravlenie*, Moscow, IN-FRA-M, 2011, 319 p.
9. **Vil van der Aalst, Kejs vann Hej.** *Upravlenie potokami rabot: modeli, metody i sistemy*, Moscow, FIZMATLIT, 2007, 316 p.

УДК 519.95

М. В. Кузнецова, аспирант,
Московский физико-технический институт, Москва,
В. В. Стрижов, д-р физ.-мат. наук, вед. науч. сотр.,
Вычислительный Центр им. А. А. Дородницына ФИЦ ИУ РАН, Москва.

Локальное прогнозирование временных рядов с использованием инвариантных преобразований*

Описан метод построения прогностической модели одномерного временного ряда. Предложено найти сегменты локальной предыстории, похожие на прогнозируемый сегмент. Близкие по заданной функции расстояния сегменты объединены в кластер. Прогностическое значение вычислено путем усреднения значений сегментов кластера. Для повышения качества прогнозирования введено инвариантное преобразование сегментов — преобразование, сохраняющее эквивалентность временных рядов на кластерах. Для преобразования использована функция, построенная с помощью метода динамического выравнивания временных рядов. Предложенный алгоритм проиллюстрирован временными рядами, описывающими движение человека и содержащими показания акселерометра. Построен ретроспективный прогноз временного ряда. В вычислительном эксперименте сравниваются два алгоритма построения прогностической модели: с кластеризацией сегментов временного ряда и с использованием метода ближайшего соседа.

Ключевые слова: кластеризация, алгоритм ближайшего соседа, путь наименьшей стоимости, динамическое выравнивание, локальное прогнозирование, монотонное преобразование, ретроспективный прогноз, опорный сегмент

* Работа выполнена при поддержке РФФИ: проект 16-07-01155.