УДК 681.3.06

Д.А. Руденко, В.А. Филатов

ФОРМАЛЬНЫЙ ПОДХОД К ОПИСАНИЮ СВОЙСТВ ДАННЫХ В ИНФОРМАЦИОННЫХ СИСТЕМАХ

Введение. Выбор формальных средств моделирования предметной области (ПО) обусловлен теми требованиями, которые предъявляются к базам данных (БД). Поскольку основным назначением БД является отражение фактов ПО, то ее формализация определяется отношениями информационных объектов как в ПО, так и в БД. При этом необходимо обеспечить единообразие представлений такого рода. Кроме того, выбранное единообразие должно обеспечить возможности реализации языка описания данных и языка манипулирования данными, при этом важным требованием, предъявляемым к языкам, является их изоморфизм при переходе от одних моделей к другим.

Разрабатываемые формальные средства основываются на идеях связи исчислений информационных объектов с операторами абстракции и импликации.

Основным этапом построения модели является фиксация категорий для анализа ПО. Прежде всего, единообразие рассуждений определяется указанием объектов данных и отношений между ними, далее для анализа различных ПО рассматривается математический спектр, представленный в форме различных категорий.

В статье исследуются формализованные модели и методы описания ПО, спецификации информационных требований пользователей, а также анализируются информационные структуры пользователей.

Рассматриваемые модели и методы обеспечивают:

- формализованное описание ПО пользователей в виде базовых компонентов и отношений между ними;
- формализованное описание спецификаций информационных требований пользователей в виде множеств отношений между информационными объектами ПО;
- анализ информационных структур локальных и распределенных БД.

Анализ литературы. Детальный анализ семантических возможностей моделей данных, проведенный в работах Х. Шмидта и Й. Свенсона [1], У. Кента [2], Б. Лангефорса [3], Д. Маклеода [4] и других авторов положил начало созданию семантических моделей БД, которые позволяют определить новые требования к функционированию ИС и моделированию ПО в целом.

По отношению к объектам существуют две проблемы: идентификация и адекватное описание объектов [5]. Для детального изучения методов идентификации объектов их синтаксического и функционального анализа можно обратиться к работам [6,7]. Дальнейшие исследования будут направлены на разрешения второй проблемы – адекватного представления объектов и определения их семантических свойств. Таким образом, материал, рассматриваемый в статье, является актуальным.

Цель проводимых исследований. Целью данной статьи является формальное описание свойств объектов ПО. Для описания свойств используются предикаты первого порядка, что позволяет абстрагироваться от модели данных информационной системы (ИС) и представить БД как множество литералов. Для полного представления об информационном содержании ИС предлагается исследовать расширенное множество информационных объектов, включающее как базовые элементы, так и вычисляемые.

Модель представления данных. Множество хранящихся данных в ИС будем рассматривать как произвольно структурированную БД. Структура БД определяется конечным набором атрибутов, выражающих свойства ПО, которые ассоциируют некоторое значение из множества допустимых значений данного атрибута с каждым объектом. Последовательность атрибутов будем называть схемой БД. При этом значения в каждом атрибуте могут изменяться, что влечет изменение состояния БД, а при этом схема остается неизменной.

В общем случае для интерпретации информации тип атрибута не имеет значения. Другими словами, если изменить тип атрибутов для каждого объекта ПО, то новое состояние должно представлять ту же информацию. По этой причине удобно предположить, что существует некоторое фиксированное множество абстрактных атрибутов, из которого выбираются атрибуты, описывающие фрагмент ПО. В качестве условных обозначений имен атрибутов будем использовать символы некоторого фиксированного алфавита, а выбранную область обозначений будем рассматривать как универсум ПО.

Для описания БД будем использовать методы логики предикатов, а при описании ПО необходимо проанализировать возможные высказывания, действующие в указанной области, и логические взаимосвязи, существующие между этими высказываниями.

Зафиксировав некоторый алфавит, можно описать взаимосвязи между атрибутами посредством использования формул. Множество сгенерированных формул будем называть аксиомами или правилами. Правила выражают тот факт, что из определенной комбинации данных в определенной БД можно вывести

набор данных, входящих в другую БД, или, что запрос будет выполнен. Семантикой БД будем называть такую интерпретацию множества правил, при котором каждое правило истинно.

Представим информационную компоненту ИС как семейство множеств:

$$D = \{D_1, D_2, \ldots, D_n\}, \quad (1)$$

где $D_i = \{d_1, d_2, \ldots, d_m\}$ – множество допустимых значений. А схему отношения как

$$R\ (A_1, A_2, \ldots, A_n), \quad (2)$$

где $A_1, A_2, \ldots, A_n$ – множества атрибутов.

Соответствие между атрибутами (2) и значениями (1) определим как отображение вида:

$$\Psi : R \rightarrow D. \quad (3)$$

Отображение (3) устанавливает, какое значение из D соответствует атрибуту из отношения R.

Таким образом, в упрощенной форме структурную компоненту ИС, представляющую БД, можно представить как $SDB = \{R, D, \Psi\}$ [8]. Как средства задания структурной компоненты могут использоваться декларативные спецификации, формулы исчисления высказываний или исчисления предикатов первого порядка. Объекты данных, которые удовлетворяют заданным условиям, составляют допустимое состояние БД.

Будем рассматривать БД как набор предикатов. В отличие от арифметических и логических функций, где область значений и область изменений аргументов по типу одна и та же, то есть однородная, у предикатов область значений функции – логическая, а область изменений аргументов – предметная. Таким образом, предикат является неоднородной функцией и может быть использован для моделирования БД.

В логике предикатов элементарным объектом, обладающим истинностным значением, является атомарная формула. Атомарная формула состоит из символического обозначения предиката и термов, выступающих в роли этого предиката. В общем виде, предикат можно представить как:

$$p(t_1, t_2, \ldots, t_n), \quad (4)$$

где p – предикат, $t_1, t_2, \ldots, t_n$ – термы.

Количество термов определяет размерность предиката, то есть в данном случае предикат p является n – местным. По сути, предикат – это функция, возвращающая булево значение «истинно» или «ложно» в зависимости от значения терма.

Аналогично (3) представим одноместный предикат $p(t)$ как отображение:

$$\varphi : p \rightarrow t. \quad (5)$$

Отображение (5) устанавливает, какое значение t должно соответствовать предикату p, чтобы формула $p(t)$ принимала значение «истинно». Тогда выражение (4) будет соответствовать одноместному предикату, а БД можно описать как множество одноместных предикатов:

$$R\ (p_1(t_1), p_2(t_2), \ldots, p_n(t_n)), \quad (6)$$

где предикат $p_i(t_i)$ ($1 \le i \le n$) принимает значение «истинно», если t_i является значением БД и «ложно» – в противном случае.

Зафиксируем некоторый алфавит $\Re$, содержащий константы, переменные и предикаты. Для одноместного предиката p формулу $p(t)$ будем называть позитивным литералом l, а формулу $\neg p(t)$ негативным литералом $\neg l$. Базисный литерал – это позитивный или негативный литерал, не содержащий переменных. Таким образом, выражение (6) можно записать как:

$$R\ (l_1, l_2, \ldots, l_n). \quad (7)$$

Ограничения целостности (ОЦ) будем выражать множеством правил

$$L = \{l \leftarrow l_1, l_2, \ldots, l_m\}, \quad (8)$$

где $l, l_1, l_2, \ldots, l_m$ – литералы ($n \ge 1$).

Двуместная логическая связка "$\neg$" представляет собой импликацию и может быть прочитана как выражение "если выполняется $l_1, l_2, \ldots, l_m$, то выполняется и l". Условие выполнимости ОЦ заключается в том, что если все литералы $l_1, l_2, \ldots, l_m$ входят в R, то и l также должен входить в R. Если такое условие не выполняется, то возможно нарушение целостности [2].

На содержательном уровне множество R представляет собой объекты ПО, а L – свойства, которым эти объекты должны удовлетворять. Основное условие правильности функционирования БД состоит в том, чтобы БД и ОЦ были совместны. Совместность заключается в отсутствии в R одного и того же позитивного и негативного литерала. Правила, которые определяют допустимые значения, задают семантику БД.

В дальнейшем исходное состояние БД будем обозначать через R, а состояние, отражающее семантику БД

через S. Например, если $R = \{a, b\}$, а $L = \{\neg b \leftarrow a\}$, то R и L совместны, а семантика $S = \{a, b, \neg b\}$ несовместна.

Под модификацией БД будем понимать операцию добавления или удаления литерала, при выполнении которой БД остается совместной. Добавление литерала означает, что l должен присутствовать в семантике модифицированной БД, а удаление – что l не должен присутствовать в семантике модифицированной БД.

Логические следствия правил. Некоторые правила выполняются во всех состояниях, в которых выполнены правила из L, будем называть такие правила следствиями. Обозначим через L^* все следствия из правил L или замыкание множества L.

Теория L–правил основывается на том, что в некотором множестве R между L–правилами существуют семантические закономерности, с помощью которых можно выводить одни правила из других, то есть делать выводы о выполнении одних правил на основании знаний, что для множества R выполняются другие правила.

Обозначим через l_i множество литералов получаемых в результате применения правил (8), а $\{l_j\}_i$ – литералы определяющие l_i. Правила (8) представим в виде:

$$L = \{l_i \leftarrow \{l_j\}_i\}, (1 \le i \le n, 1 \le j \le m). \quad (9)$$

Будем говорить, что множество S удовлетворяет правилам (9) если все элементы l_i входят в S, то есть $l_i \subseteq S$. Рассмотрим два экстремальных вида правил:

$$\varnothing \leftarrow \{l_j\}_1 \quad (10)$$

и

$$l_i \leftarrow \varnothing. \quad (11)$$

Правило (10) тривиально удовлетворяет любому L. Правило (11) удовлетворяет такому S, в котором все элементы $l_i \in S$. В дальнейшем такие правила рассматриваться не будут.

Для множества S в любой момент существует некоторое множество правил L, которым это множество удовлетворяет. Пусть задано два множества S_1 и S_2 и пусть L удовлетворяет S_1 и не удовлетворяет S_2. Необходимо выявить все допустимые правила из L (в обозначении L'), удовлетворяющие S_1 и S_2 (или показать отсутствие такого набора правил).

Чтобы найти L', необходимы семантические знания о S_1. Эти знания определяются множеством L, так как правила являются первичными по отношению к БД и по существу задают ограничения на объекты БД. Замыкание правил L^*, применимых к множеству R конечно, так как существует конечное число подмножеств множества R. Таким образом, всегда можно найти все правила L, которые удовлетворяют S_1, перебрав все возможные правила.

Однако такой подход требует больших временных затрат. Если известны некоторые правила $L \in L'$, то можно вывести остальные правила. Множество правил L влечет за собой правило $l_i \leftarrow \{l_j\}_i$ если все объекты из S, удовлетворяющие всем правилам из L, также удовлетворяют правилам $l_i \leftarrow \{l_j\}_i$. Вывод правил – это процедура устанавливающая, что если S удовлетворяет определенным правилам, то оно должно удовлетворять и некоторым другим правилам, не входящим в L.

Определим правила вывода.

1. Рефлексивность. Если $l \in R$, то $l \leftarrow l$.

Множество R всегда имеет хотя бы один элемент, следовательно, $l \leftarrow l$ имеет место в R.

2. Аддитивность. Если $l_1 \leftarrow l$ и $l_2 \leftarrow l$, то $l_1, l_2 \leftarrow l$.

Так как множество правил определяют семантику S, то есть свойства порожденных литералов, то из $l_1 \leftarrow l$ и $l_2 \leftarrow l$, то l_1 следует, что l_1 и l_2 имеют одинаковые свойства, определяемые литералом l. Из этого следует, что $l_1, l_2 \leftarrow l$ удовлетворяет семантики S.

3. Транзитивность. Если $l \leftarrow l_1$ и $l_1 \leftarrow l_2$, то $l \leftarrow l_2$.

Порождаемые литералы являются результатом применения правил к определяющим литералам и последующее применение этих правил к порожденным литералам. Следовательно, литерал l является результатом применения правил, определяющих последовательную зависимость литералов. Следовательно, $l \leftarrow l_2$ удовлетворяет S.

4. Пополнение. Если $l \leftarrow l_1$, то $l \leftarrow l_1, l_2$.

Так как для порождения литерала l достаточно применить заданные операции к базовому литералу l_1, то правило вида $l \leftarrow l_1, l_2$, включающее дополнительный базовый литерал l_2 также имеет место для порождения l, хотя при этом является избыточным.

5. Псевдотранзитивность. Если $l_1 \leftarrow l_2$, $l \leftarrow l_1, l_3$, то $l \leftarrow l_2, l_3$.

Доказательство этой аксиомы вытекает из аксиом рефлексивности, пополнения, аддитивности и транзитивности. Если $l \leftarrow l_1, l_3$ и $l_1 \leftarrow l_2$, то вследствие аксиомы рефлексивности имеем $l_3 \leftarrow l_3$. Согласно аксиоме пополнения $l_1 \leftarrow l_2, l_3$ и $l_3 \leftarrow l_2, l_3$. По аксиоме аддитивности имеем $l_1, l_3 \neg l_2, l_3$. Применив аксиому транзитивности, получим $l \leftarrow l_2, l_3$.

6. Проективность. Если $l_1, l_2 \leftarrow l$, то $l_1 \leftarrow l$ и $l_2 \leftarrow l$.

Аксиома 6 в некоторой степени обратна аксиоме 2. Так как литералы l_1 и l_2 имеет одинаковые свойства, определяемые литералом l и это правило удовлетворяет S, то в правилах $l_1 \leftarrow l$ и $l_2 \leftarrow l$ литералы l_1 и l_2 также будут иметь эти же свойства, удовлетворяющие S.

Очевидно, что $L \subseteq L^*$ и что $L^{**} = L^*$. Два множества S_1 и S_2 логически эквивалентны (в обозначении $S_1 \equiv S_2$), если $L_1^* = L_2^*$. Но построение L^* соответствует перебору всех подмножеств множества L, что занимает экспоненциальное время [9].

Одним из способов уменьшения времени проверки вхождения литерала в S является построение замыкания множества R относительно правил L. Замыканием множества R называется такое множество литералов R^*, для которых $l^* \leftarrow l \in L^*$ и не существует ни одного литерала из R, который бы зависел от l и не принадлежал l^*. Известно, что получение замыкания является трудоемкой задачей для аналитических вычислений. Поэтому рассмотрим алгоритм, позволяющий автоматизировать этот процесс для правил, не обладающих взаимозависимыми и циклическими свойствами.

Вычисление замыкания R: Вход. R, $L = \{l_i \leftarrow \{l_j\}_i\}$. Выход. R^*.

В общем виде алгоритм вычисления замыкания R может быть представлен в следующем виде. Будем использовать дополнительную переменную M для сохранения множества литералов. Пусть $M := R$. Последовательно пересматривая правые части правил l_i, проверяем условие $l_i \in R$. Если условие $\{l_j\}_i \subseteq R$ выполняется, то модифицируем $M := R \cup \{l_j\}_i$, исключаем $l_i \leftarrow \{l_j\}_i$ из L и продолжаем пересматривать правые части правил, начиная с первого правила в модифицированном множестве L. Если не найдено не одного правила из правой части $\{l_j\}_i$, для которой $\{l_j\}_i \subseteq R$, алгоритм закончен.

Для того чтобы убедиться в эквивалентности семантик двух множеств R_1 и R_2 достаточно построить замыкание для одного из них и проверить вхождения в замыкание каждого элемента второго множества. Таким образом, условие $S_1 \equiv S_2$ справедливо, если $R_1 \subseteq R_2^*$ и $R_2 \subseteq R_1^*$.

Выводы. Временная сложность рассмотренных алгоритмов вычисления замыканий зависит от размера входного множества L. Таким образом, меньшее множество правил гарантирует более быстрое исполнение этих алгоритмов.

Анализ требований к информационным системам показывает, что, как правило, L содержит достаточно большое количество правил, которые значительно замедляют работу алгоритма. Такая ситуация дает повод для разработки более эффективных методов определения эквивалентных БД. В частности, выделить набор базисных правил, исключив тривиальные и избыточные правила и таким образом уменьшить размер L.

Полученные результаты подтверждают адекватность предлагаемых метода и алгоритма и могут быть использованы при усовершенствовании технологий доступа и обработки данных, а также реализации методов представления БД для процедур вывода результатов запроса средствами реляционных систем управления данными. Дальнейшие исследования в рассматриваемой области необходимо сконцентрировать на вопросах организации запросов в системах БД с неоднородной структурой, и разработки методов построения и поддержки запросов для гетерогенных информационных систем и распределенных баз данных.

ЛИТЕРАТУРА

1. Schmid, H.A., Swenson, J.R. On the semantics of the relation model // In: Proc. of ACM SIGMOD Int. Conf. Management of Data. – 1975. – P. 211 – 223.
2. Kent, W. Consequences of assuming a universal relation. – ACM Trans. on Database Systems. – 1981. – V. 3. – P. 3 –17.
3. Langefors, B. Information systems //. Information Processing 74. – Amsterdam: North-Holland. – 1974. – P. 937 – 945.
4. McLeod, D. The semantic data model. – MIT Press. – 1979.
5. Цаленко, М.Ш. Моделирование семантики в базах данных. – М.: Наука. Гл. ред.. физ.-мат. лит., 1989. – 288с.
6. Kent, W. Limitations on record-based information models. – ACM Trans. on Database Systems. – 1979. – V. 4. – P. 107 – 131.
7. Langefors, B. Infological models and information user views // Inform. Systems. – 1980. – V. 5 – P. 17 – 32.
8. Буслік, М.М. Оптимальні зображення реляційних баз даних. // Монографія. – К.: ІСДО, 1993. – 84 с.
9. Схрейвер, А.А. Теория линейного и цело численного программирования. Т. 1 / Пер. с англ. – М.: Мир, 1991. – 360 с.

ФИЛАТОВ Валентин Александрович – доктор технических наук, профессор кафедры искусственного интеллекта Харьковского национального университета радиоэлектроники

Научные интересы: базы данных и знаний, агентные технологии, мультиагентные системы, извлечение знаний из данных.

РУДЕНКО Диана Александровна – кандидат технических наук, доцент кафедры информатики Харьковского национального университета радиоэлектроники

Научные интересы: модели данных, базы данных и знаний, распределенные информационные системы.