

## ИНТЕГРАЦИЯ ИНФОРМАЦИОННЫХ СИСТЕМ ПРИ ЧАСТИЧНОМ ОТОБРАЖЕНИИ МОДЕЛЕЙ ДАННЫХ

*Эффективность информационных систем определяется не столько их производительностью, сколько способностью к расширению, в частности, устойчивостью к изменениям в течение жизненного цикла системы, удобством развития и поддержания работоспособности. В статье рассмотрены методы построения отображений моделей данных различной структуры и определены этапы построения частичного отображения, включающего отображения схем и состояний баз данных. Такой подход позволяет использовать модели, отображающие связи отдельных интегрируемых баз данных, что обеспечивает динамичность развития информационной системы в целом.*

*Ефективність інформаційних систем визначається не стільки їхньою продуктивністю, скільки здатністю до розширення, зокрема, стійкістю до змін протягом життєвого циклу системи, зручністю розвитку та підтримки працездатності. У статті розглянуто методи побудови відображень моделей даних різної структури і визначено етапи побудови часткового відображення, що включає відображення схем і станів баз даних. Такий підхід дозволяє використовувати моделі, які відображають зв'язки між базами даних що інтегруються, та забезпечувати динамічність розвитку інформаційної системи в цілому.*

*Efficiency of information systems is defined not so much by their productivity, how many ability to expansion, in particular, stability to changes during life cycle of system, convenience of development and maintenance of working capacity. In article methods of construction of displays of models of data of various structures are considered and stages of construction of the partial display including displays of schemes and conditions of databases are certain. Such approach allows using models displaying communications of separate integrated databases that provides dynamism of development of information system as a whole.*

### **Введение**

Представление данных в виде алгебраических структур является естественным для многих прикладных задач. Начиная с работы Е. Кодда [1], применение алгебраических методов распространилось на моделирование баз данных (БД). Объектом изучения стали алгебраические системы с операциями над отношениями, среди которых выделяются реляционные и цилиндрические алгебры. Авторами, исследовавшими эти объекты, являются Н. Andreka [2], I. Nemeti [3], F. Vanchilhon [4], Б.И. Плоткин [5] и др.

Вместе с тем центральной проблемой создания автоматизированных систем остается разработка прикладных программ, обеспечивающих решение конкретных задач в сфере действия системы. При этом эффективность систем определяется не столько их производительностью, сколько способностью к эволюции, в частности, устойчивостью к изменениям в течение жизненного цикла системы, удобством развития систем и поддержания их работоспособности.

Прикладные программы оказываются существенно зависимыми от конкретных систем управления базами данных (СУБД) и используемых в них структур данных. Мобильность прикладных программ по отношению к различным типам СУБД является достаточно низкой. Существенно, что используемые при этом языковые средства должны быть такими, чтобы полученные спецификации можно было применить к любой СУБД. По мере развития систем возникают задачи, требующие одновременного использования информации, накопленной в нескольких (возможно, различных) БД, что при существующей организации СУБД трудно реализуемо. Кроме этого, особое место занимает проблема управления распределенными БД организованными средствами разнотипных СУБД с неоднородной структурой данных.

В связи с интенсивным развитием техники распределенной обработки данных на основе компьютерных сетей, проблема интегрированного использования децентрализованной совокупности БД требует эффективного решения.

Проблемы использования распределенных автоматизированных систем совместной обработки данных рассматривались в работах В.М. Глушкова [6], Г.С. Пospelова [7], М.Ш. Цаленко [8], Л.А. Калиниченко [9] и др.

На основании проведенного анализа интегрированные БД можно определить как комплекс алгебраических и языковых средств, ориентированный на разработку прикладных программ, независимых от СУБД при одновременном взаимодействии с несколькими, возможно, неоднородными БД. Следует отметить, что интеграция должна строиться таким образом, чтобы обеспечивалась возможность эволюционного развития системы, выраженного в допущении автономного использования, включая модификацию интегрируемых БД в рамках локальных прикладных программ одновременно с их использованием в составе единого интегрированного пространства.

Особую нишу при интеграции занимают неоднородные БД, в которых различия можно трактовать двояко. С одной стороны, БД, реализованные средствами различных СУБД, являются структурно неоднородными по соответствующим им моделям данных. С другой стороны, БД, поддерживаемые одной и той же СУБД, но определяемые различными концептуальными схемами, являются информационно неоднородными. В дальнейшем в понятие неоднородных БД будем включать оба вида неоднородности.

Интеграцию неоднородных систем можно реализовывать несколькими способами, в частности, посредством консолидации данных, распределения фрагментов данных по узлам и построения системы на основе федерализации данных. В основе решения задачи консолидации и федерализации данных лежит вопрос преобразования структур и/или моделей данных, что требует приведения неоднородных структур и моделей к единому виду. Вопрос эффективного использования межмодельных отображений на сегодняшний день является актуальным и рассматривается как обобщенная методология интеграции неоднородных баз данных.

Таким образом, целью статьи является исследование и модификация алгебраических средств моделирования БД, а также обоснование построения и использования частичных отображений между моделями БД неоднородной структуры.

### 1. Отображение моделей данных

Центральной проблемой интеграции БД является построение отображений соответствующих моделей. Модель данных будем представлять как кортеж вида

$$M^i = \langle D^i, A^i, S^i, O^i \rangle,$$

где  $D^i$  – множество допустимых значений,  $A^i$  – множество имен допустимых значений из  $D^i$ ,  $S^i$  – множество схем БД, определяющихся отношениями между значениями из  $D^i$  и  $O^i$  – операционная спецификация на отношении  $R^i$ .

Информационным объектом рассматриваемой предметной области будем называть отображение вида  $\alpha: A^i \rightarrow D^i$ . Таким образом, схему  $S^i$  будем рассматривать как некоторое отношение между информационными объектами. Множество допустимых состояний, соответствующих некоторой схеме  $S_i \in S^i$ , представляют собой множество функций  $V^{S_i}: A^{S_i} \rightarrow D^i$ , определяющих для каждого объекта, объявленного в схеме и обозначенного некоторым идентификатором  $A_i \in A^i$ , его значение  $D_i \in D^i$ . Пространство состояний, выразимое средствами модели  $M^i$ , представляет собой множество функций  $V^i: A^i \rightarrow D^i$ , рассматриваемое как объединение множеств  $V^i = \bigcup_{S_i \in S^i} V^{S_i}$ .

Отображение модели данных  $M^j$  в модель данных  $M^i$  в обозначении  $\rho: M^j \rightarrow M^i$  представляет собой совокупность отображений [9]:

- пространства состояний БД  $M^j$  в пространство состояний БД  $M^i$  –  $\eta: V^j \rightarrow V^i$ ;
- схем БД  $M^j$  – в схемы БД  $M^i$  –  $\varphi: S^j \rightarrow S^i$ ;
- операционной спецификации  $M^j$  – в последовательность операций  $M^i$  –

$\psi: O^i \rightarrow P^j$ , где  $P^j$  – процедура на языке модели  $M^j$ .

Двум моделям  $M^i$  и  $M^j$  могут соответствовать различные отображения  $\rho$ , отличающиеся видом функций  $\eta$ ,  $\varphi$ , и  $\psi$ . Говорят, что отображение модели  $M^j$  в модель  $M^i$  правильное, если оно удовлетворяет критериям:

- определенности, то есть произвольное состояние БД модели  $M^i$  представимо в  $M^j$ ;
- интерпретируемости, то есть любой оператор  $O^j$  модели данных  $M^j$  имеет интерпретацию в модели  $M^i$ ;
- согласованности, то есть любое изменение БД в модели  $M^j$  воспроизводимо средствами  $M^i$ .

При рассмотрении правильности отображений моделей данных существуют некоторые затруднения, возникающих при построении  $\rho$ .

В дальнейшем операционную спецификацию будем рассматривать как объединение  $O^i = O^{S_i} \cup O^{V_i}$ , где  $O^{S_i}: S^i \rightarrow V^i$  – семантическая функция определения данных в  $M^i$ , а  $O^{V_i}$  – семантическая функция манипулирования данными в  $M^i$ .

Действие функций  $\varphi$ ,  $\eta$ ,  $O^{S_i}$ ,  $O^{S_j}$  можно отобразить с помощью коммутативной диаграммы, представленной на рисунке 1.

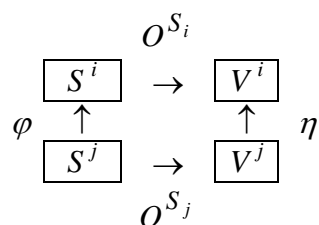


Рисунок 1. Диаграмма определенности отображения моделей.

С учетом того, что функции являются гомоморфизмами, то есть  $O^{S_i} \circ \varphi = \eta \circ O^{S_j}$ , то диаграмма является коммутативной. С другой стороны, из коммутативности приведенной диаграммы еще не следует, что пространство всевозможных состояний должно совпадать с отображением  $\eta$  состояний  $V^j$  модели  $M^j$  в состояния  $V^i$ .

Для  $V^j: A^j \rightarrow D^j$  множество элементов данных, составляющих область определения, представляет собой множество типов элементов данных  $T_{S^j} = \{\tau_1^j, \tau_2^j, \dots, \tau_n^j\}$ , определенных в схеме  $S^j$ . Подобным образом  $V^i$  соответствует множество типов данных  $T_{S^i} = \{\tau_1^i, \tau_2^i, \dots, \tau_m^i\}$ . Состояние интегрированной БД определяется допустимой комбинацией типов  $\tau_1^i, \dots, \tau_1^j, \dots$  в схеме, полученной на основании  $S^j$  и  $S^i$ . Отображение  $\eta$  является тривиальным, если  $|\tau^j| = |\tau^i|$ , то есть  $n = m$ . Такое отображение реализуемо, если модели  $M^j$  и  $M^i$  структурно подобны.

При различии  $M^j$  и  $M^i$  воспользуемся разбиением исходных множеств типов данных  $T_{S^j}$  и  $T_{S^i}$  на подмножества

$$C_{T_{S^j}} = \{c_1^j, c_2^j, \dots, c_p^j\} \text{ и } C_{T_{S^i}} = \{c_1^i, c_2^i, \dots, c_k^i\},$$

где  $c_p^j = \{\tau_1^j, \dots, \tau_v^j\} \subset T_{S^j}$  и  $c_k^i = \{\tau_1^i, \dots, \tau_w^i\} \subset T_{S^i}$ .

Рассмотрим три возможных случая для множеств  $C_{T_{S^j}}$  и  $C_{T_{S^i}}$ .

1.  $C_{T_{S^j}} \cap C_{T_{S^i}} = \emptyset$ ;
2.  $C_{T_{S^j}} \cap C_{T_{S^i}} \neq \emptyset$ ;

$$3. C_{T_{S^j}} \cap C_{T_{S^i}} = C_{T_{S^j}} (C_{T_{S^i}}).$$

Очевидно, что случай 3 представляет собой тривиальное отображение  $\eta$ . Для случаев 1 и 2 типы данных  $c_p^j$  должны выбираться таким образом, чтобы, с одной стороны, в их состав были включены типы данных, между которыми в схеме  $S^j$  заданы логические зависимости, соответствующие модели  $M^j$ , а с другой, чтобы  $c_p^j$  можно было бы отобразить в  $c_k^i$  без потери информации. Таким образом, каждому компоненту  $V^j$  ставится в соответствие совокупность  $C_{T_{S^j}}(S^j)$ , которая посредством  $\eta$  отображается в совокупность  $C_{T_{S^i}}(S^i)$ , при этом  $S^i = \varphi(S^j)$ .

Критерию определенности отображения  $M^j$  в  $M^i$  можно поставить в соответствие инъективное отображение  $\eta$ . Различным состояниям БД в  $M^j$  соответствуют в этом случае различные состояния БД в  $M^i$ , отражающие существенную информацию, представленную в модели  $M^j$ . Однако, если  $\eta$  инъективно, то в  $M^i$  могут допускаться и другие состояния БД, которым в  $M^j$  не соответствует какой-либо образ. Отсюда следует, что одним из критериев корректности отображения  $M^j$  в  $M^i$  является коммутативность диаграммы отображения схем, в которой отображение  $\eta$  должно быть биективным. Свойства отображения  $\eta$  и  $\varphi$  означают следующее:

- состояние БД в  $M^j$  интерпретируемо средствами  $M^i$ ;
- схема БД в  $M^j$  отображаема в схему БД в  $M^i$ .

## 2. Условия коммутативности частичного отображения моделей данных

При отображении моделей  $M^j$  в  $M^i$  могут возникнуть два принципиальных затруднения. Во-первых, как было отмечено, коммутативность диаграммы отображения схем БД и биекция функции  $\eta$  определяют тот факт, что композиция функции  $O^{S^i} \circ \varphi$  определяет пространство состояний БД модели  $M^i$ , эквивалентное пространству состояний БД модели  $M^j$ . При отображении модели  $M^j$  в некоторую другую модель, например, в  $M^k$ , означает, что в композиции должна измениться только функция отображения схем  $\varphi$  (обозначим эту функцию  $\varphi_k$ ), но с другой стороны, композиция  $O^{S^i} \circ \varphi_k$  должна также обеспечивать получению в  $M^i$  пространства состояний, эквивалентное состояниям в  $M^j$ . Другими словами, семантика модели  $M^i$  должна быть достаточно гибкой для того, чтобы ее средствами можно было бы выразить произвольные логические зависимости данных, соответствующие различным моделям данных. Таким образом, для построения коммутативной диаграммы отображения схем необходимо иметь средство учета полного набора зависимостей между данными.

Во-вторых, из свойства инвариантности функции отображения схем  $\varphi$  не следует инвариантность операционной спецификации  $\psi$ . То есть, это означает, что различным внутренним структурам модели  $M^j$  могут соответствовать различные требования к семантике операционной спецификации модели  $M^i$ .

Рассмотренные проблемы указывают на невозможность отображения произвольной модели данных  $M^j$  в произвольную модель  $M^i$  без изменения семантики данных и операционной спецификации. Традиционно считается, что для корректного отображения  $M^j$  в  $M^i$  необходимо построение новой модели  $M^{ij}$  на основе модели  $M^i$  так, чтобы она отражала логические зависимости модели  $M^j$ .

Модель  $M^{ij}$  называют интерпретацией модели  $M^j$  в модель  $M^i$ , которая включает основные компоненты моделей данных: множество схем БД  $S^{ij}$ , выразимых в модели  $M^{ij}$ , семантику определения данных  $O^{S^{ij}} : S^{ij} \rightarrow V^{ij}$ , определяющих состояние БД, операционную спецификацию  $O^{ij}$  и семантику операторов  $O^{V^{ij}}$  [9].

Рассмотрим интерпретацию частичного отображения  $M^j$  в  $M^i$ . Для построения

соответствующей интерпретации будем использовать две компоненты: множество схем  $S^{ij}$  и функцию определения данных  $O^{S_{ij}}$ .

Расширение модели  $M^i$  до модели  $M^{ij}$  может быть реализовано специальными языковыми средствами, которые обеспечивают задание в терминах  $M^i$  логических зависимостей данных, соответствующих модели  $M^j$ , и задание семантики операторов определения данных  $O^{S_{ij}}$ . Интерпретацию логических зависимостей данных при расширении  $M^i$  будем определять аксиоматически посредством языка логики предикатов первого порядка. Таким образом, при расширении модели  $M^i$  будем использовать единую систему аксиом  $LD^i$ , в соответствии с которыми формируется модель  $M^{ij}$ , выражающая зависимости данных ( $L^i$  - правила) как в терминах модели  $M^i$ , так и в терминах модели  $M^j$  [10].

При построении расширения  $M^i$  до  $M^{ij}$  необходимо решить, каким образом пользователь модели  $M^{ij}$  получит информацию о дополнительных логических зависимостях данных. Один способ заключается в задании семантики операторов манипулирования данными в виде соответствующих процедур [11]. Другой способ заключается в сравнении зависимостей  $L^i$ , выведенных на основании аксиом  $LD^i$  - системы и построении соответствующего покрытия зависимостей для использования его в модели  $M^{ij}$ . В рассматриваемом подходе будем отдавать предпочтение аксиоматическому способу построению модели  $M^{ij}$ .

Расширение модели  $M^i$  будет образовываться путем дополнения набора правил, входящих в модель, правилами, выведенными на основе системы аксиом  $LD^i$ . В дальнейшем структуру модели данных будем представлять в виде:

$$M^i = \langle D^i, A^i, S^i, O^i, LD^i \rangle,$$

где  $LD^i$  - система аксиом, выражающих логические зависимости данных модели  $M^j$  в терминах модели  $M^i$ .

Отображением  $f$  модели данных  $M^j$  в расширение  $M^{ij}$  модели  $M^i$  будем называть совокупность отображений:

- пространства состояний БД  $M^j$  в пространство состояний БД расширенной модели  $M^{ij}$  -  $\bar{\eta} : V^j \rightarrow V^{ij}$ ;
- схем БД  $M^j$  - в схемы БД расширенной модели  $M^{ij}$  -  $\bar{\varphi} : S^j \rightarrow S^i \cup \bar{A}^i$ , где  $\bar{A}^i$  - множество имен значений, выведенных на основании системы аксиом  $LD^i$ .

Рассматривая частичное отображение  $M^j$  в  $M^i$ , определим условия коммутативности диаграмм. Отображение  $f = \langle \bar{\eta}, \bar{\varphi} \rangle$  модели  $M^j$  в расширение  $M^{ij}$  модели  $M^i$  является коммутативным, если выполняются условия, представленные на диаграмме рисунка 2.

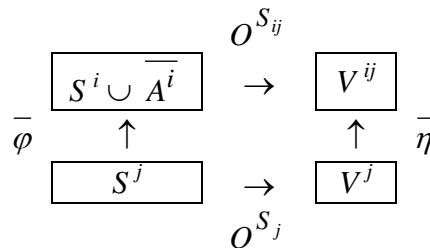


Рисунок 2. Диаграмма отображения схем и пространства состояний.

Критерий коммутативности отображений является основным критерием, по которому устанавливается возможность интеграции произвольной модели данных  $M^j$  с моделью  $M^i$ . Модель данных  $M^j$  интегрируема с моделью  $M^i$ , если существует расширение  $M^{ij}$  модели  $M^i$ , обеспечивающее коммутативное отображение  $f : M^j \rightarrow M^{ij}$ . Для обеспечения коммутативного отображения необходимым условием является эквивалентность расширения  $M^{ij}$  модели  $M^j$ .

Прежде чем определить эквивалентность моделей, определим эквивалентность состояний БД. Состояния БД в моделях  $M^j$  и  $M^{ij}$  будем называть эквивалентными  $V^j \equiv V^{ij}$ , если они соответствуют одному и тому же состоянию в рамках абстрактного, независимого от конкретных моделей представлению, обеспечивающему однозначность семантических свойств моделей и определяющих их сходства при использовании единого языка. Будем предполагать, что при частичном отображении выполнение таких условий также достаточно, чтобы состояния БД были эквивалентны и отражали одну и ту же совокупность фактов.

Модели данных  $M^j$  и  $M^i$  будем называть эквивалентными  $M^j \equiv M^i$ , если каждой схеме БД  $s_j \in S^j$  в  $M^j$  можно поставить в соответствие схему БД  $s_i \in S^i$  в  $M^i$  (в обозначении  $s_j \Rightarrow s_i$ ) и при этом справедливо обратное утверждение, то есть,  $s_j \Leftrightarrow s_i$ , и каждому состоянию  $V^j$  схемы  $s_j$  можно поставить в соответствие эквивалентное состояние  $V^i$  схемы  $s_i$  (в обозначении  $V^j \Rightarrow V^i$ ), при этом обратное утверждение также справедливо, то есть,  $V^j \Leftrightarrow V^i$ .

При рассмотрении неоднородных структур данных и в общем случае моделей данных необходимо учитывать различие в правилах построения схем БД. В таких случаях анализ только правил, определяющих логические зависимости между данными, недостаточен. Для обеспечения полного соответствия схем в множество правил будем включать также структурные правила, определяющие составные типы объектов схемы. Введение структурных правил необходимо для того, чтобы истинность аксиом была инвариантной по отношению к операциям модификации данных из  $O^i$  [12].

Аксиома  $l^i$  является инвариантом модели  $M^i$  по отношению к изменению схемы этой модели, если из истинности  $l^i(T_{S^i})$  следует истинность  $l^i(o_i(T_{S^i}))$  для любых  $o_i \in O^i$  и произвольных типов данных  $T_{S^i}$  схемы  $s_i$ . Система аксиом  $LD^{ij}$  расширения  $M^{ij}$  модели  $M^i$  логически полна, если отображению  $M^j$  в  $M^{ij}$  соответствует коммутативная диаграмма отображения схем БД. С другой стороны, допущение изменения схем локальных БД усложняет логические зависимости данных при расширении  $M^i$  до  $M^{ij}$ . Таким образом, необходимо рассмотреть вопрос о структурной полноте системы аксиом  $LD^{ij}$ .

Система аксиом  $LD^{ij}$  расширения  $M^{ij}$  модели  $M^i$  структурно полна, если любое действие по изменению схемы БД модели  $M^i$  соответствует некоторой аксиоме  $l^{ij} \in LD^{ij}$ , либо предусмотрено соответствующим оператором  $o_j \in O^j$ , модели данных  $M^j$ .

Полная (как логически, так и структурно) система аксиом  $LD^{ij}$  образует полную схему инвариантов расширения  $M^{ij}$  модели  $M^i$ . Очевидно, что полная схема инвариантов определяет систему условий непротиворечивости БД. Следует также отметить, что в коммутативном отображении модели  $M^j$  в расширение  $M^{ij}$  модели  $M^i$  система аксиом  $LD^{ij}$  должна представлять собой полную схему инвариантов.

### 3. Одностороннее отображение схем баз данных

Реализация построения отображения  $M^j$  в расширение  $M^{ij}$  достаточно сложная алгоритмическая задача, так как необходимо обеспечить коммутативность диаграмм схем БД и пространства состояний при биективном отображении инвариантов  $M^{ij}$ . Для эффективного решения этой задачи предлагается декомпозировать ее на последовательность независимых подзадач, решения которых не должны быть зависимы от вида исходных моделей  $M^i$  и  $M^j$ .

Независимость процедур обеспечивается, если использовать единый метод определения семантики данных в терминах одной и той же метамодели, для чего можно воспользоваться моделью, описанной в [13].

Основными этапами построения неполного коммутативного отображения  $M^j$  в  $M^i$  являются:

- 1) формальное определение моделей данных  $M^j$  и  $M^i$  средствами приведенной модели;

- 2) построение отображения  $M^j$  в  $M^i$  при инъективном отображении  $\eta: V^j \rightarrow V^i$ , включая проверку коммутативности диаграммы отображения схем;
- 3) расширение модели  $M^i$  до  $M^{ij}$ , включая определение правил и формальную проверку полноты  $LD^{ij}$ ;
- 4) построение отображения  $M^j$  в  $M^{ij}$ , включая проверку коммутативности диаграмм отображения схем.

При этом, если на каком-либо этапе полученная диаграмма не коммутативна, то необходимо проанализировать один из предыдущих этапов и внести изменения в построенные ранее отображения. Рассмотренный подход к построению отображения моделей может являться основой для приведения некоторых моделей (включая неоднородные) к виду, удовлетворяющему критериям интеграции БД.

Говоря об интеграции нескольких разнородных моделей, можно сделать вывод о том, что общее для всех моделей расширение  $\bar{M}$  является объединением расширений всех интегрируемых моделей. То есть, если принять, что интегрируется  $n$  БД и, соответственно, рассматривается  $n$  моделей, то общее представление данных будет выглядеть как объединение  $\bar{M} = \bigcup_{\substack{i=1, \dots, n-1 \\ i=i+1, \dots, n \\ i < j}} M^{ij}$ , при этом, очевидно, что количество всех

возможных объединяемых пар будет соответствовать  $C_n^2 = \frac{n(n-1)}{2}$ .

Применение метода коммутативного отображения допускает интеграцию только эквивалентных моделей или их подмножеств, то есть необходимо выполнение условия  $M^j \subseteq \bar{M}$ . Это в свою очередь означает, что интеграция невозможна, если логические зависимости интегрируемых БД (а точнее их замыкание  $L^+$ ) не пересекаются.

Реализация поддержки отображений может осуществляться как средствами конкретной СУБД, так и средствами прикладной программы, обеспечивающей передачу данных. Использование средств СУБД может быть эффективным, если модель и структура локальных БД не изменяется в процессе функционирования интегрированной системы. Понятие независимости прикладных программ от СУБД означает, что при их разработке используется такая модель данных  $M^{jn}$ , что для реализации БД может быть использована произвольная СУБД с моделью данных  $M^j$ , поддерживающая коммутативное преобразование модели данных  $M^j$  в  $M^{jn}$ . Решение проблемы независимости от СУБД предполагает фиксацию такой модели  $M^{jn}$  для каждой прикладной программы и построение отображения  $M^{jn}$  в модель  $M^i$ .

В отличие от проблемы интеграции БД средствами расширения моделей и построения обобщенной схемы, использование концепции независимости прикладных программ означает, что проектирование прикладных систем и соответствующих БД развивается от предметной области через независимые схемы к БД конкретной СУБД.

Учитывая соответствующую последовательность при проектировании интегрированной системы и вытекающую отсюда целесообразность и возможность скрыть реализацию объектов конкретной БД, вполне достаточно применить одностороннее отображение модели данных. Диаграмма отображения схем БД при одностороннем отображении будет иметь вид, представленный на рисунке 3.

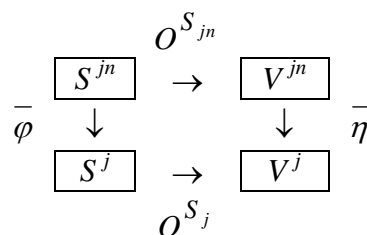


Рисунок 3. Диаграмма одностороннего отображения схем.

В таком отображении  $\bar{\varphi}$  является инъективным, так как в локальных БД могут быть определены схемы, которым в интегрированной системе нельзя поставить в соответствие какую-либо схему  $s_{jn} \in S^{jn}$ . Отображение  $\bar{\eta}$  также теряет свойство биективности, так как для схемы  $s_j \in S^j$  (а также в случае  $s_j = \sigma(s_{jn})$ ) могут допускаться состояния  $v_j \in V^j$ , которым нельзя поставить в соответствие состояние  $v_{jn}$ , допустимое в схеме  $s_{jn}$ .

При работе с БД для достижения коммутативности диаграммы схем и обеспечения одностороннего отображения необходимо придерживаться следующих правил:

- 1) в  $S^j$  включаются только те  $s_j$ , которые имеют единственный прообраз отображения  $\varphi: S^j \rightarrow S^i$ ;
- 2) модификация БД средствами  $M^j$  организуется таким образом, что для всех состояний  $v_j \in V^j$  ставится в соответствие единственное состояние  $v_{jn} \in V^{jn}$ ;
- 3) использование средств манипулирования данными, соответствующих  $M^j$  недопустимо, то есть реализация операций над данными в  $M^{jn}$  должна выполняться средствами, определенными в интегрируемых моделях.

Рассмотренные правила отображения обеспечивают возможность независимого функционирования локальных БД. В интегрируемой системе модель  $M^{jn}$  рассматривается как подмножество  $M^{jn} \subseteq M^j$ . При этом, если существует дополнение  $M^j \setminus M^{jn}$ , то оно представляет инварианты БД, отражающие особенности конкретной модели. С другой стороны, так как предполагается возможность расширения модели  $M^j$ , то  $M^{jn}$  также является расширяемой моделью, что обеспечивает гибкость функционирования интегрированной системы в целом.

### **Выводы**

В статье рассмотрена задача частичного отображения произвольной модели данных  $M^j$  в  $M^i$ , возникающая при интеграции неоднородных баз данных. Рассмотрен метод отображения моделей, реализуемый аксиоматически посредством системы инвариантов  $LD^{ij}$ , определяющей ограничение множества допустимых состояний БД в  $M^i$  по отношению к модели  $M^j$ . Определено, что для интеграции различных моделей, отображения схем и соответствующие им пространства состояний должны быть коммутативными. Описана последовательность этапов построения частичных коммутативных отображений произвольных моделей данных, базирующаяся на использовании общей системы аксиом  $LD^{ij}$ .

Для построения независимых от СУБД прикладных программ поддержки корректности отображений при допущении модификации структур БД введена модель  $M^{jn}$ , соответствующая каждой локальной прикладной программе. Построено инъективное отображение схем БД и определена последовательность шагов для достижения коммутативности схем БД при одностороннем отображении моделей данных.

Рассмотренный подход к построению отображения  $M^j$  в  $M^i$  является основой построения программных компонент, реализующих такие отображения в системе интеграции неоднородных БД. Построение коммутативных отображений моделей данных неоднородных БД вместе с системой аксиом позволяет выявить эквивалентность предметных областей в различных моделях данных, а при необходимости установить степень различия таких моделей.

Дальнейшее развитие систем интеграции неоднородных моделей данных при допущении локальных структурных изменений требует расширения языковых средств, соответствующих операций и организации контроля целостности данных. Коммутативность отображений позволяет определить необходимые условия интеграции, но для практического построения корректно функционирующей интегрированной системы необходимо учитывать особенности функционирования локальных систем и иметь возможность контролировать работу системы в целом. То есть, необходимо

определить дополнительные условия, допускающие независимость моделей  $M^{jn}$  от модификаций в моделях  $M^j$  и  $M^i$ .

#### ЛИТЕРАТУРА

1. Codd E.F. A relational model for large shared data banks. *Comm. of ACM* 13, 6, 1970, p.377-387.
2. Andreka H., Nemeti I. Applications of universal algebra, model theory and categories in computer science (Part 1) *Comput. Lingust. Comput. Lang.*, 13, 1979, p. 152-282.
3. Andreka H., Nemeti I. Applications of universal algebra, model theory and categories in computer science (Part 2) *Comput. Lingust. Comput. Lang.*, 14, 1980, p. 43-65.
4. Banchilhon F. On the Compliteness of Query Language for Relational Data Bases. *Lect. Notes in Comp. Sci.*, V.64, Springer-Verlag, 1978, pp.76-98.
5. Плоткин Б.И. Универсальная алгебра, алгебраическая логика и базы данных. М.:Наука, 1991. – 448 с.
6. Глушков В.М. и др. Сети ЭВМ. – М.: Связь. 1977. – 280 с.
7. Поспелов Д. А. Введение в теорию вычислительных систем. – М.: Советское радио, –1972. – 184 с.
8. Цаленко М.Ш. Семантические и математические модели баз данных. – М.: ВИНТИ. – 1985. – 207 с.
9. Калиниченко Л.А. Методы и средства интеграции неоднородных баз данных. - М.: Наука, 1983. - 423 с.
10. Танянский С.С. Модель согласования данных при интеграции информационных систем. // *Радиоэлектроника и информатика – Науково – технічний журнал.* - № 2 – Харків, ХНУРЕ, 2006. с. 84-91.
11. Мальков Ю.А. Метод обработки неоднородных данных в задачах интеграции информационных систем. // *Тезисы докладов научно-практической конференции Информатизация высших учебных заведений МВД Украины.* – НУВД. – Харьков. – 2007.-с. 12.
12. Танянский С.С. Семантическая модель предметной области в задачах интеграции неоднородных информационных систем. // *Вестник Херсонского национального технического университета.* – Херсон. 2005. - №1(21). с. 52-59.
13. Пономаренко Л. А. Програмні агентні технології в адмініструванні баз даних / Л. А. Пономаренко, В. О. Філатов. // *Вісник Київського торговельно-економічного університету.* – 2001. – №3. – С. 68–73.