

конструкций, характерных для языков высокого уровня. Семантический подход при reengineering'e состоит в применении базы знаний (БЗ) для соответствующих преобразований. База знаний может быть представлена в форме трехуровневой модели знаний (грамматика – таблица решений – исчисление предикатов) [2]. Верхним уровнем БЗ являются правила принятия решений, выраженные в форме Хорновских дизъюнктов. Для приведенного выше примера в БЗ можно включить правила формирования оператора цикла и условного оператора в виде:

а) Оператор Цикла ($f(i, m_0, m_1, m_2, n_0, n_1, n_2, a)$):
 $Init(n_0, i, m)$, $Body(n_1, n_2, i, a)$, $Сmp(n_2, i, m_1)$,
 $Inc(n_2, i, m_2)$, $Jmp(n_2, n_1)$.

б) Условный Оператор ($f(i, m_2, n_0, n_1, n_2, a)$):
 $Сmp(n_0, i, m_2)$, $Jmp(n_0, n_2)$, $Body(n_1, n_2, a)$.

Оператор цикла состоит из блоков инициализации некоторой рабочей переменной ($Init$), сравнения ($Сmp$), увеличения (Inc), передачи управления (Jmp) и тела цикла ($Body$). Условный оператор включает в себя конструкции сравнения, передачи управления и тело оператора. Обозначения:

– $f(b_1, \dots, b_k)$ – некоторая функция от k переменных;
 – n_0, n_1, n_2 – номера блоков (соответственно – начальный, промежуточный, конечный);

УДК 519.62

ДОПОЛНИТЕЛЬНЫЕ КОМПОНЕНТЫ ИНФОРМАЦИОННОЙ МОДЕЛИ ГИС

ЛЯХОВЕЦ С.В., ЧЕТВЕРИКОВ Г.Г.

Приводятся дополнительные компоненты информационной модели ГИС. Эти компоненты предназначены для решения проблемы управления проектами сложных природно-технических комплексов. Рассматриваются компоненты для хранения топологических данных, иерархии трубопроводов, произвольной группировки событий, а также все необходимые таблицы баз данных для хранения данных компонентов модели.

Введение

На протяжении жизненного цикла любого трубопровода из различных источников собираются большие количества информации в разных форматах. Возникает необходимость создания информационной модели для организации и управления данными, процессами сбора и использования информации. Информационная модель трубопровода – это согласованная, хорошо управляемая, легко доступная совокупность всей информации о трубопроводе, среде и эксплуатационной истории. Ее независимость от конкретных программ является существенным фактором для обеспечения гибкости и возможности многократного использования од-

– i – рабочая переменная;

– m_0, m_1, m_2 – значения рабочей переменной (начальное, величина изменения, конечное);

– a – тело цикла или условного оператора.

Литература: 1. *Водолажский А.С.* Преобразование программ в удобную для логического анализа форму // Сборник научных трудов по материалам 4-го Международного молодежного форума “Радиоэлектроника и молодежь в XXI веке”. Х.: ХНУРЭ, 2000. С. 333-334. 2. *Дюбко Г.Ф., Валенда Н.А., Водолажский А.С.* Семантический подход к автоматическому анализу программ / Сборник научных трудов по материалам 5-й Международной конференции “Теория и техника передачи, приема и обработки информации”. Х.: ХНУРЭ, 1999. С. 360-362.

Поступила в редколлегию 25.03.2002

Рецензент: д-р техн. наук, проф. Авраменко В.П

Водолажский Алексей Сергеевич, студент ХНУРЭ. Научные интересы: защита информации. Адрес: Украина, 61115, Харьков, ул. 2-Пятилетки, 2-В, кв. 144, тел. 94-71-29.

Дюбко Геннадий Федорович, канд. техн. наук, профессор кафедры ПО ЭВМ ХНУРЭ. Научные интересы: формальные системы. Адрес: Украина, 61140, Харьков, пр. Гагарина, 38, кв. 70, тел. сл. 40-94-46, дом. 27-13-26.

них и тех же данных. Нарботки в этой области можно изучить в источниках [15, 16].

Информационная модель предоставляет следующие преимущества:

1. Качественное использование и доступ к данным: уменьшение избыточности данных; многократное использование данных; оптимизация связей между отделениями компании.

2. Делает более эффективным принятие решений: постепенное сокращение работы при соборе и форматировании данных; расширенная аналитическая среда; улучшение формулировки рационального решения, построения сценария и тестирования; улучшение инструментальных средств документирования; способность включать различные варианты в среду принятия решения.

3. Поддерживает приложения для улучшения эксплуатационной эффективности: поддержка отчетов и разрешений; оперативное получение информации по району работ; автоматизированная картография – создание подписей объектов и полосы карты непосредственно из базы данных; планирование и маршрутизация для службы и запросов технического обслуживания; генерация карт размещения, дающих ориентировочную информацию для облегчения обнаружения находящегося под землей оборудования; интеграция с документацией на оборудование позволяет обеспечивать доступ к новейшим документам по техническому обслуживанию, действующему оборудованию и методикам.

4. Позволяет применять приложения оценки риска, что обеспечивает поддержку работоспособности оборудования.

5. Улучшает конкурентоспособность: повышает надежность и обслуживание – технологии ГИС (Географические Информационные Системы) [10-14] позволяют сокращать время обнаружения отказа оборудования, определять место и посылать команды обслуживания, обеспечивать их достоверной информацией; анализ поставок; анализ связей; маршрутизация продукта – ГИС определяет маршрут продукта.

При рассмотрении разнообразных источников и типов данных, которые входят в информационную модель, видно, что никакая отдельная технология не в состоянии делать управление, анализ и отображение информации. Для реализации информационной модели требуется использование набора технологий. Такой подход применяет конкретную технологию для управления соответствующими данными, при этом разработанная технология и архитектура данных облегчает интеграцию данных в информационной модели.

Реляционные системы управления базами данных [1-9]. Корпоративная реляционная база данных и основанные на локальной сети ведомственные реляционные базы данных обеспечивают первичное хранение данных, внесенных в информационную модель [17-19]. Эти таблицы баз данных должны иметь ключ, который связывает данные с определенным объектом. ГИС играет первостепенную роль в информационной модели, как технология интеграции, основанная на объединении данных исключительно по географической привязке. ГИС выполняет функции менеджера данных, предоставляет приложениям среду для поддержки автоматической картографии или выполнения сложного пространственного моделирования, обеспечивает механизм привязки для разнородных данных, хранящихся в реляционной базе данных. Документы, рисунки, данные обследования и другие данные могут быть привязаны к географическому положе-

нию. ГИС дает возможность работать с разнообразными пространственными видами объектов и позволяет получать более подробные данные для конкретного объекта.

Для описания особенностей объектов в информационной модели трубопровода необходимы компоненты для хранения его топологических данных, иерархии трубопроводов и компоненты, которые позволят произвольную группировку событий.

Пояснения к описанию материала

Модель линейной привязки позволяет хранить в реляционной базе данных (РБД) пространственные данные, расположенные вдоль центральной линии трубы. Модель данных позиционирования представляет собой большой набор связанных между собой таблиц РБД. Далее опишем предназначенную для этого структуру данных.

Сначала сделаем пояснение к рисункам.

Если на таблице нарисована стрелочка ↗, это значит, что в таблице соответствующее поле, где находится стрелка, имеет ссылку на себя. Например, для нахождения предыдущего события (Previous_Event_ID) → События. Связь между таблицами изображена стрелками →, которые показывают, по какому полю таблицы идет связь. В первой строке таблицы написано ее название на английском и русском языках. Первая колонка таблицы – это оригинальное название ее полей. Вторая колонка несет информацию о ссылочном типе поля. Если в соответствующей ячейке стоит pk (primary key), то это главный ключ таблицы, если fk (foreign key) – это поле может использоваться для ссылки на другие таблицы. В третьей колонке указан тип данных, содержащихся в этом поле. Внизу каждой таблицы описаны необходимые индексы для организации ссылок и поиска.

Во всех таблицах модели, имеющих поле с именем, которое заканчивается на “_CL” (Code List), хранится ссылка на таблицу, поля которой, как правило (если не указано особо), имеют одинаковое

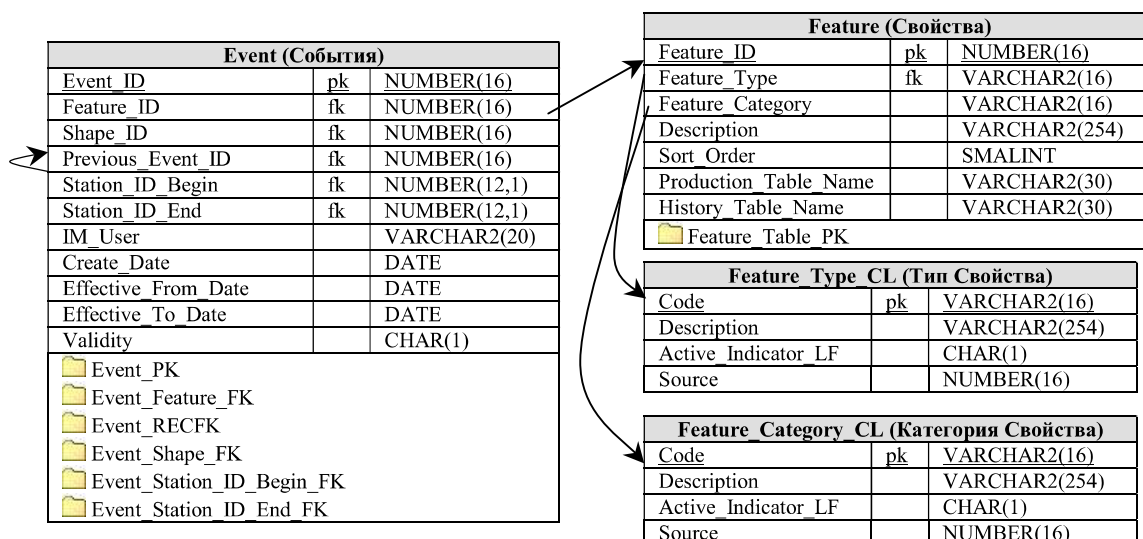


Рис.1. Структура таблиц События и Свойства

название. Но таблицы называются по-разному (например, Line_Type_CL). Эти таблицы определяют заданное фиксированное количество значений, они определены в модели.

В таблицах встречаются поля, ссылающиеся на таблицы *Источник* и *Единицы Измерения*. Первая таблица предназначена для хранения информации об источнике, из которого поступили данные. Вторая — определяет единицы измерения, которыми пользовались при определении параметров.

Ядро модели — компонент Событие

Рассмотрим сначала ядро информационной модели. Ядром концепции информационной модели является таблица *Событие* (рис. 1). Из нее выходят все связи событий в модели. События — это объекты, хранимые в модели, например трубы, отчеты, результаты обследований и др. Каждому событию, входящему в модель, соответствует Event_ID — уникальный номер, который присваивается в то время, когда добавляется новая запись в таблицу *Событий*. Каждая запись в таблице уникальна и имеет свой набор атрибутов. Для определения, в какой таблице находятся атрибуты данного события, необходимо из таблицы *Свойства* по значению поля Feature_ID, находящемуся в таблице *События*, выбрать соответствующую запись. По значению Production_Table_Name (это имя таблицы) в этой записи можно определить, в какой таблице искать атрибуты. И потом по идентификатору события (Event_ID) в *Атрибутивных* таблицах можно извлечь необходимые значения атрибутов.

Разберем подробно значение всех полей таблиц.

Table Event — основная таблица. Используется для сохранения идентификаторов *Атрибутивных* таблиц.

Event_ID — основной ключ таблицы (pk). Непрограммируемое число, используемое для связи с таблицами *Атрибутов*. Каждая таблица атрибутов имеет ключевое поле Event_ID, и по нему, после определения имени атрибутивной таблицы, нужно выбрать конкретные атрибуты.

Feature_ID — указатель на таблицу *Свойств* (Feature). В таблице *Свойства* определяется таблица *Атрибутов*, которая используется для хранения подробностей данного события.

Shape_ID — указатель на таблицу *Форма* (Shape). Создает связь с координатами x, y, z без использования привязки к месту по центральной линии трубопровода, т.е. ссылается на абсолютные координаты. (Эта часть модели пока неразработана).

Previous_Event_ID — указатель на запись события, которая была заменена этим событием. Используется для ведения истории.

Station_ID_Begin — связь с таблицей позиционирования. Хранит идентификатор начальной точки для линейных событий. Позиционирование использу-

ется для привязки событий к центральной линии трубопровода.

Station_ID_End — связь с таблицей позиционирования. Хранит идентификатор конечной точки для линейных событий.

IM_User — идентификатор пользователя. Требуется для ведения контрольного журнала.

Create_Date — дата создания записи события.

Effective_From_Date — первая дата, когда данное событие было в действии. Может быть использовано для планирования реализации блока событий.

Effective_To_Date — последняя дата, когда данное событие было в действии.

Validity — флаг, используемый приложениями. Идентифицирует подтверждение того, что данные имеют силу.

Таблица Свойства

Таблица *Свойства* связана с таблицей *События* и определяет тип и местонахождение атрибутов, зависящих от конкретного события. Это важная таблица в информационной модели. Она позволяет приложениям узнавать о существовании таблиц *Атрибутов*. Поэтому существует возможность создавать новые таблицы и добавлять их в информационную модель в любое время. Например, когда добавляется новая *Атрибутивная* таблица, она идентифицируется в таблице *Свойства* (добавляется новая запись в таблицу, поле в которой Production_Table_Name содержит имя новой таблицы). Все совместимые приложения немедленно осведомляются о новой таблице. Необходимо заметить, что каждая запись в таблице *Событий* содержит Feature_ID для соединения с таблицей *Свойства*.

Table Feature — определяет тип и местоположение атрибутов событий.

Feature_ID — основной ключ таблицы, внешний ключ (fk) для возврата в таблицу *Событий*. Позволяет выбрать из таблицы *События* все события, имеющие соответствующий тип.

Feature_Type — используется для группировки линий, точек и других элементов.

Feature_Category — используется для группировки *Атрибутивных* таблиц. Может быть использован для группировки событий по событиям *Оборудование*, *Обследование* или по другим категориям событий.

Description — используется для хранения длинного описания таблицы атрибутов события.

Sort_Order — элементарная возможность сортировки. Используется для генерации отчетов. Будет расширяться в будущем для добавления большого количества сценариев сортировки.

Production_Table_Name — хранит местоположение (имя таблицы) хранения связей атрибутов и свойств.

History_Table_Name – хранит местоположение (имя таблицы), где содержатся записи истории о том, когда над таблицей *Атрибутов* были проведены операции модификации и удаления. Это позволяет сохранять историю за пределами экземпляра базы данных.

Таблица *Свойства* необходима в информационной модели для предотвращения последовательного извлечения данных приложениями из таблицы *Событие*. Таблица *Свойства*, связанная с таблицей *Событий*, становится картой навигации, идентифицирующей, где хранятся данные в модели. Новые таблицы могут быть добавлены в любой промежуток времени.

Имя таблицы атрибутов и свойств (Production_Table_Name) и таблицы *История* (History_Table_Name) хранятся в таблице *Свойства*. Они используются как указатели на соответствующие таблицы. Через поле Production_Table_Name осуществляется доступ ко всем таблицам *Атрибутов*, как имеющихся в модели, так и добавленных в нее.

Идентификатор предыдущего события (Previous_Event_ID) в таблице *Событие* и History_Table_Name в таблице *Свойства* используется для хранения исторических данных. Он позволяет применять два подхода для хранения данных истории. Первый подход (использующий Previous_Event_ID) позволяет редактировать набор данных в той же таблице *Событие* и создает новый Event_ID для каждого редактирования или изменения, сделанного с записью события. Получается как бы цепочка одного и того же события в разных модификациях, пройдя по которой, можно вернуться к исходному событию. Для нахождения предыдущего события в цепочке необходимо по значению Previous_Event_ID в текущем событии найти предыдущее событие в таблице *События*. При этом подходе таблица *Атрибутов* значительно вырастает в размере.

Для создания больших атрибутивных таблиц предложили данные истории хранить во вторичном наборе данных, чтобы он был как второй экземп-

ляр, с названием структур идентичных тем, которые используются в основных таблицах модели. Таким образом, могут существовать экземпляры основной базы данных и *Истории* с идентичной структурой данных. Экземпляр основной базы данных будет содержать только одну текущую версию события. Экземпляр *Истории* будет содержать все предыдущие версии события. Доступ к основной базе данных осуществляется через Production_Table_Name, а к базе данных истории – через History_Table_Name.

Компонент Сеть

Таблица *Сеть* обеспечивает средствами для хранения топологических данных, необходимых для программ гидравлического моделирования. Эти программы основаны на векторно-топологическом представлении данных. В векторной топологии дуги – это линейные свойства, такие как сегмент трубы. Программам моделирования требуется добавлять новую дугу сегмента, если изменяется диаметр или материал трубы. Запись в таблице *Сегмент_Сети* (Network_Segment) связана с событием в таблице *Сегмент_Трубы* (Pipe_Segment). Узлы (junctions) – это конечные точки дуг свойств. Узел находится в точке встречи двух или более дуг, каждый конец которых висящий (не подсоединен). Заметим, что узлы не обязательно связаны с участком трубопроводного оборудования; узлы – это абстрактный класс данных.

Каждая гидравлическая модель требует своей собственной сети. Предлагаемая структура (рис.2) приспособляется для хранения многочисленных, потенциально перекрывающихся сетей. Опишем структуру подробно.

Network_Segment – таблица для хранения свойств дуг.

Event_ID – основной ключ таблицы, внешний ключ для обратной связи с таблицей *Событий*. Эта таблица не хранит сегмент сети, который не представлен в другом месте модели как событие, т.е. сегмент сети должен быть описан в таблице *События*, а это поле указывает его идентификационный номер.

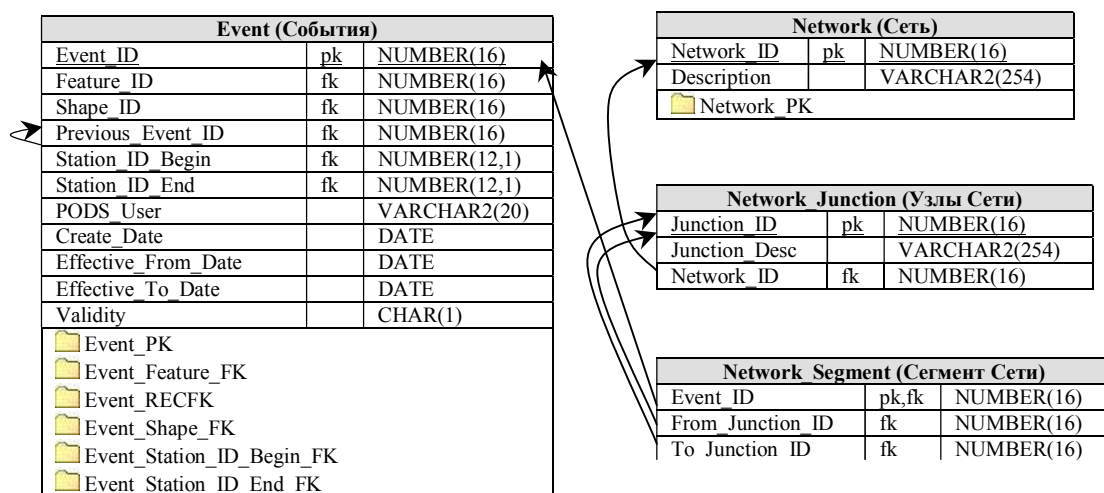


Рис.2. Структура таблиц *Сети*.

From_Junction_ID – идентификатор узла, от которого начинается сегмент дуги. Узел пространственно соответствует первой вершине в сегменте дуги. From_Junction_ID – это внешний ключ к таблице Узлы_Сети (Network_Junction), в которой хранятся узлы.

To_Junction_ID – идентификатор узла, в котором заканчивается сегмент дуги. Узел пространственно соответствует последней вершине в сегменте дуги. To_Junction_ID – это внешний ключ к таблице Узлы_Сети.

Network_Junction – таблица для хранения свойств узлов.

Junction_ID – уникальный идентификатор конкретного узла. Это бессмысленный уникальный номер.

Junction_Desc – опционально. Текстовое описание узла.

Network_ID – идентификатор сети, которой принадлежит узел.

Network – таблица для хранения сети.

Network_ID – уникальный идентификатор сети. Это бессмысленный уникальный номер.

Network_Desc – опционально. Текстовое описание сети.

Компонент Иерархия

Таблица *Иерархия* предоставляет средства для создания произвольных, иерархических групп трубопроводов. Эти таблицы (рис.3) предназначены в основном для использования с ветвящимися, собирающимися или расходящимися системами, помогают облегчать общие балансовые подсчеты. Они могут использоваться для облегчения расчетов получения разности фактического/расчетного количества в передающих системах.

Иерархии могут быть созданы с любым желаемым количеством уровней. Можно включать таблицы с бесконечной степенью вложенности. К тому же дочерние записи (ветвь в структуре дерева) могут иметь более одного родителя, так могут быть учтены ветви линий и циклов. Так как иерархии наследуются произвольно, предлагаемые таблицы поддерживают множественные иерархии.

Хотя предлагаемые таблицы *Иерархии* реализовывают линейную уровневую модель, они могут

реализовать и последовательную модель для трубопроводных серий.

Опишем структуру подробно.

Connection – таблица для хранения линейных иерархических связей.

Connection_ID – уникальный идентификатор определенного соединения ветви. Это не обязательно соответствует физической ветви в трубопроводной системе; иерархия может быть произвольна.

Child_Line_ID – идентификатор линии текущего иерархического уровня. Это внешний ключ к таблице *Линии* (Line). Таблица *Линии* относится к разделу в модели, который называется позиционирование.

Parent_Line_ID – идентификатор родительской линии в следующем, более высоком уровне иерархии. Таблица *Линии* относится к разделу в модели, который называется позиционирование.

Hierarchy_ID – идентификатор иерархии, куда входит это соединение.

Hierarchy – таблица для хранения линейной иерархии.

Hierarchy_ID – уникальный идентификатор узла иерархии.

Description – текстовое описание иерархии.

Компонент Группировки

Концепция *Групп* – это мощное средство. Она дает возможность одному или более событиям быть зависимыми в иерархическом виде. Группы используются для описания местоположения, подающих систем и комплексного оборудования, такого как компрессорные станции и перерабатывающие заводы. Часть модели, отвечающая за группы, состоит из двух таблиц (рис.4): *Группа Событий* (Event Group) и *Группа Событий_Ссылки* (Event_Group_Cross_Ref).

Рассмотрим таблицы подробнее.

Event_Group – таблица предназначена для хранения имени группы и родительской группы. Нужно отметить, что *Группа Событий* – это не специальная таблица событий. *Группа Событий* имеет рекурсивные связи, поэтому возможно иметь одну группу событий, входящую в другую.

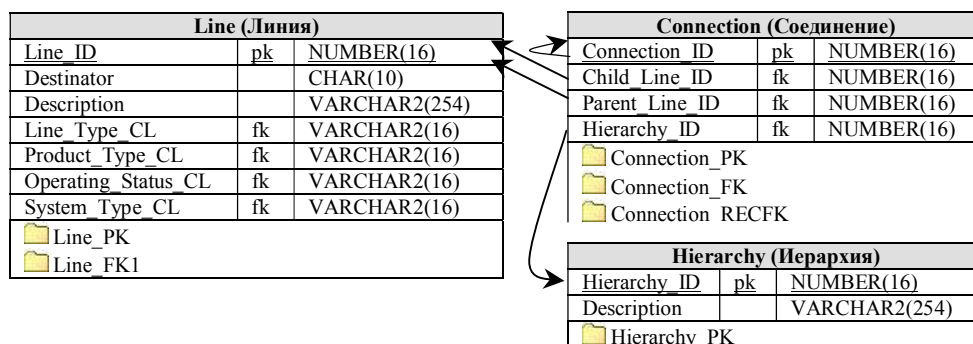


Рис.3. Структура таблиц *Иерархии*

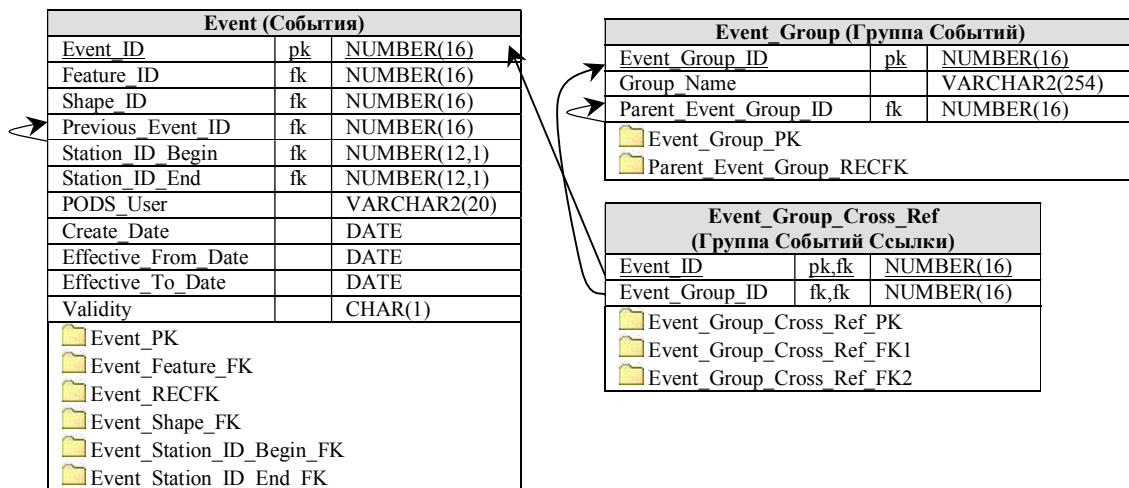


Рис.4. Структура таблиц Группировки.

Event_Group_ID – уникальный идентификатор группы событий.

Group_Name – текстовое название группы событий.

Parent_Event_Group_ID – поле хранит идентификатор родительской группы для данной группы событий.

Event_Group_Cross_Ref – таблица хранит перекрестные ссылки между событиями и группами событий. Она связывает таблицу *Событий* с таблицей *Группа Событий*. Любое событие может быть включено в любую группу. События могут также быть включены во множество групп. Каждая группа может иметь одного или более членов группы. Записи членов группы в действительности являются указателями на другие записи атрибутов события.

Event_ID – идентификатор события, входящего в группу событий.

Event_Group_ID – идентификатор группы событий, в которую входит данное событие.

Выводы

Информационная модель, предназначенная для организации и управления данными, процессами сбора и использования информации, является большим шагом вперед в области информационной поддержки трубопроводов. В данной статье рассмотрена часть модели, предназначенная для описания особенностей объектов в информационной модели трубопровода. Рассмотрены компоненты для хранения топологических данных трубопровода, иерархии трубопроводов и компоненты, которые позволяют производить произвольную группировку событий.

Литература: 1. *Дейт К.* Введение в системы баз данных. М.: Наука, 1980. 464 с. 2. *Компьютерные технологии обработки информации* / Под ред. С.В. Назарова. М.: Финансы и статистика, 1995. 247 с. 3. *Мартин Г.* SQL: Справочное руководство. М.: Лори, 1997. 291 с. 4. *Мартин Дж.* Организация данных в вычислительных

системах. М.: Мир, 1978. 662 с. 5. *Орлов В.Н., Лантнев В.С.* Модели данных в СУБД / Под ред. В.И. Першикова. М.: МО СССР, 1982. 124 с. 6. *Системы управления базами данных и знаний: Справочное издание* / А.Н. Наумов, А.М. Вендров, В.К. Иванов и др.; Под. ред. А.Н. Наумова. М.: Финансы и статистика, 1991. 352 с. 8. *Ульман Дж.* Основы систем баз данных. М.: Финансы и статистика, 1983. 334 с. 9. *Date C.J., Hugh D.* Foundation for Object/Relational Databases: The Third Manifesto. Reading, Mass.: Addison-Wesley, 1998. 152с. 10. *Коновалова Н.В., Капранов Е.Г.* Введение в ГИС. М.: Мир, 1997. 160с. 11. *Цветков В.Я.* Геоинформационные системы и технологии. М.: Финансы и статистика, 1998. 288с. 12. *С.Ю. Желтов и др.* Особенности реализации 3D ГИС / Информационный бюллетень ГИС-Ассоциации). 1997. №5(12). С.12-14. 13. *Mladen Stojic.* 3-D GIS: unleash the power // GEOEurope. 2000. №11. С.30-33. 14. *Кошкарёв А.В., Тихунов В.С.* Геоинформатика. М.: "Картгеоцентр" – "Геодиздат", 1993. 213 с. 15. *Integrated Spatial Analysis Techniques - Topical Report no. GRI-97/0072 / Gas Research Institute.* 1997. 328p. 16. *The OpenGIS Abstract Specification, OpenGIS Project Document Number99-100r1 // Open GIS Consortium.* 1999. 443p. 17. *Бойко В.В., Савинков В.М.* Проектирование информационной базы автоматизированной системы на основе СУБД. М.: Финансы и статистика. 1982. 174 с. 18. *Вейнеров О.М., Самохвалов Э.Н.* Проектирование баз данных САПР. М.: Высш. шк., 1990. 144 с. 19. *Пржиялковский В.В.* Абстракции в проектировании баз данных // СУБД. 1998. №1-2. С. 90-97.

Поступила в редколлегию 12.02.2002

Рецензент: д-р техн. наук, проф. Алексеев О.П.

Ляховец Сергей Витальевич, аспирант кафедры ПО ЭВМ ХТУРЭ. Научные интересы: геоинформационные системы, модели данных, технологии визуализации данных. Хобби: программирование, компьютеры, туризм, спелеология. Адрес: Украина, 61166, Харьков, просп. Ленина, 14, тел. ((380)-0572)-409-446.

Четвериков Григорий Григорьевич, канд. техн. наук, доцент кафедры ПО ЭВМ ХТУРЭ. Научные интересы: разработка теории и практика использования методов синтеза многозначных структур языковых систем искусственного интеллекта. Адрес: Украина, 61166, Харьков, просп. Ленина, 14, тел. ((380)-0572)-409-446, ((380)-0572)-279-748.