

УДК 004.652: 004.655: 004.623

В.И. Есин

Харьковский национальный университет им. В.Н. Каразина, Харьков

РЕИНЖИНИРИНГ СУЩЕСТВУЮЩИХ БАЗ ДАННЫХ

Описывается проблема необходимости реинжиниринга существующих баз данных информационных систем компаний. Решить данную проблему предлагается с помощью метода, опирающегося на использование базы данных с универсальной моделью.

Ключевые слова: реинжиниринг, универсальная модель данных, база данных, база данных с универсальной моделью данных.

Необходимость реинжиниринга существующих баз данных

Сегодня можно говорить, что эра, когда разработки информационных систем приходили в организацию и начинали проекты информатизации «с нуля», прошла. Большинство организаций уже имеет некоторые информационные системы (ИС), которые со временем становятся бременем компании и начинают требовать реинжиниринга. Это объясняется тем, что специалистам любой крупной и давно существующей компании, которая обладает несколькими базами данных, относящимися к разным видам деятельности, данные в которых могут иметь разные представления и быть даже несогласованными, становится очень трудно связывать и анализировать содержащуюся в них информацию. К тому же многие базы данных (БД), составляющие основу таких ИС, как правило, построены на уже устаревших «платформах» (например, Dbase, FoxPro и т.д.). При этом данные, в них хранящиеся, имеют большую практическую ценность.

Как результат, имея самые разнообразные изолированные друг от друга источники данных, найти необходимые для деятельности компании данные становится невозможным. Поэтому чтобы упростить и ускорить доступ к такой информации, дать возможность проследить связи между разными источниками данных и обеспечить надежность использования унаследованных настольных баз данных в обеспечении управления на всех уровнях, необходимо каким-то образом интегрировать эту информацию.

Заниматься реорганизацией информационных систем не является для разработчика БД чем-то необычным. Сегодня организации обычно расходуют от 20 до 40% своего ИТ-бюджета на эволюционирование своих данных путем миграции (изменение местоположения данных), преобразования (изменения формы или структуры данных) или очистки (изменение или повторный ввод данных для последующего использования) [1].

А раз подобные задачи встраивания устаревших информационных компонентов в систему, основанную на новой технологии, сегодня достаточно часто прихо-

дится решать, то нужно сделать так, чтобы они были разрешимыми. То есть, чтобы компоненты унаследованных систем сохраняли интероперабельность. Потому что, чем дольше живет и приносит пользу информационная система, тем это выгоднее для компании. Естественно, что для этого должна существовать возможность добавления в нее компонентов, спроектированных и разработанных, вообще говоря, в другой технологии. Сегодня в мире существует большое количество подходов, методов и технологических решений, напрямую или косвенно соотносимых с деятельностью по реинжинирингу ИС. Однако они не интегрированы на уровне методологий (процессов разработки). Как результат, можно наблюдать наличие огромного их количества, где основной акцент сделан на опять же разработку ИС «с нуля», и практическое отсутствие «стройных» методологий, целью создания которых являлось бы комплексное, целостное решение задач реинжиниринга ИС [2].

Ниже предлагается метод (подход), с помощью которого решается эта проблема.

Реинжиниринг существующих баз данных информационных систем в схему базы данных с универсальной моделью

Метод основывается на использовании при построении информационных систем базы данных с универсальной моделью данных [3].

Определим основные этапы метода и дадим краткую их характеристику.

1. *Экспорт данных во временные таблицы.* Данные базы данных, подвергающейся реинжинирингу, экспортируются в одну из СУБД: Oracle, PostgreSQL или Access.

2. *Создание таблицы-шаблона.* В той СУБД, куда были экспортированы данные, создается так называемая таблица-шаблон, с атрибутами, представленными в заголовке табл. 1. Кратко охарактеризуем эти атрибуты.

Атрибут «*преобразуемая таблица*» (столбец 2) содержит имя временной таблицы (в одной из СУБД: Oracle, PostgreSQL или Access), в которую были перемещены данные из БД, подвергающейся реинжини-

рингу. Столбец 3 – «атрибут» – содержит имя атрибута во временной таблице, в которую были перемещены данные из БД, подвергающейся реинжинирингу. Атрибут «оператор ЯМД» (столбец 4) содержит оператор (точнее определенную его конструкцию) языка модели данных (ЯМД) [6], которому может быть присвоено имя атрибута столбца 3 соответствующей строки табл. 1. В столбце 5 – атрибут – «оператор РазделВлад» указывается или конструкция ЯМД <Раздел>=, или последовательность операторов ЯМД (иерархия разделов ПрО – элемент строки метаописания) обязательно заканчивающаяся конструкцией <Раздел>=. В столбце 6 – атрибут – «Раздел Влад» указывается имя раздела, присваиваемое конструкции <Раздел>= (столбец 5). Оно может быть константой, адресом поля временной таблицы БД, адресом собственно БД с УМД, в которую переносятся данные (последние впоследствии при необходимости можно уничтожить). В столбце 7 – атрибут – «оператор ЭкзОВлад» указывается или конструкция ЯМД <ЭкзО>=, или последовательность операторов ЯМД (иерархия экземпляров объектов ПрО – элемент строки метаописания), обязательно заканчивающаяся конструкцией <ЭкзО>=. В столбце 8 – атрибут – «ЭкзОВлад» указывается имя экземпляра объекта, которое присваивается конструкции <ЭкзО>= (столбец 7), и которое может быть константой, адресом поля временной таблицы БД, адресом собственно БД с УМД, в которую переносятся данные. В столбце 9 – атрибут – «оператор КлассОЭкзОВл» указывается конструкция ЯМД <КлассО>=, или последовательность операторов ЯМД (иерархия классов объектов ПрО – элемент строки метаописания), обязательно заканчивающаяся конструкцией <КлассО>=. В столбце 10 – атрибут – «КлассОЭкзОВлад» указывается имя класса объекта-владельца, которое присваивается конструкции <КлассО>= (столбец 9), и которое может быть константой, адресом поля временной таблицы БД, адресом собственно БД с УМД, в которую переносятся данные. В столбце 11 – атрибут – «оператор ТипОЭкзОВл» указывается или конструкция ЯМД <ТипО>=, или последовательность операторов ЯМД (элемент строки метаописания), обязательно заканчивающаяся конструкцией <ТипО>=. В столбце 12 – атрибут – «ТипОЭкзОВл» указывается имя типа объекта-владельца, которое присваивается конструкции <ТипО>= (столбец 11), и которое может быть константой, адресом поля временной таблицы БД, адресом собственно БД с УМД, в которую переносятся данные. В столбце 13 – атрибут – «оператор I» указывается или какой-то конкретный оператор ЯМД, или последовательность требуемых операторов ЯМД (элемент строки метаописания). В столбце 14 – атрибут – «Значение I» указывается имя, которое присваивается конструкции (столбец 13), и которое может быть константой, адресом поля временной таблицы БД, адресом собственно БД с УМД, в которую переносятся данные. Атрибутов подобных указанным в столбцах 13 и 14

табл. 1 может быть несколько (до 12 пар), в зависимости от необходимости моделирования элементов рассматриваемой предметной области.

Таблица-шаблон может быть создана, например, и в офисном приложении Microsoft Excel (либо ему подобном), а потом экспортироваться в одну из рекомендуемых СУБД.

3. *Заполнение (формирование) таблицы-шаблона.* С использованием различных конструкций и операторов языка модели данных [4], заполняется таблица-шаблон (табл. 1).

При составлении конструкций, помещаемых в таблицу-шаблон, следует учитывать особенности их формирования. А именно те, которые связаны с добавлением определенных служебных символов в стандартные конструкции операторов строк метаописаний ЯМД. Рассмотрим их более подробно.

Символ \$, указанный первым в строке атрибутов, начинающихся словом «оператор» таблицы-шаблона (столбцы 5, 7, 9, 11, 13), свидетельствует о том, что информация, обрабатываемой строки, используется для формирования строки метаописания. При этом за символом \$ обязательно должен следовать необходимый оператор ЯМД (так как любые другие символы будут игнорироваться) и последним служебным словом в этой строке метаописания должно быть одно из следующих: <КлассО>, <КлассС>, <ЭкзО>, <Папка>, <Документ>.

Символ \$, указанный первым в строке атрибутов, начинающихся словом «значение» таблицы-шаблона, свидетельствует о том, что информация о значении, обрабатываемой строки, будет выбираться из БД с УМД. При этом следует помнить, что, обращаясь к БД с УМД по сформированной таким образом строке метаописания, возможно, произойдет выбор нескольких значений. А так как это не всегда является допустимым, то пользоваться подобным приемом следует очень аккуратно.

Кроме того, считывать из БД с УМД можно только данные (метаданные – нельзя)!

Символ #, указанный первым в строке атрибутов, начинающихся словом «значение», или указанный внутри строки метаописания, определенного символом \$ в строке атрибутов, начинающихся словом «оператор» таблицы-шаблона, свидетельствует о том, что данные выбираются из соответствующего атрибута (имя атрибута определяется идентификатором, следующим за символом #) указанной таблицы БД, которая преобразуется.

4. *Реинжиниринг исходной базы данных в БД с УМД.* С помощью специального приложения, на основании содержания строк и столбцов таблицы-шаблона, на языке модели данных формируются и исполняются строки метаописания. В результате исполнения приложением строк метаописаний ЯМД происходит запись в схему БД с УМД метаданных и данных рассматриваемой ПрО, ранее хранящихся в исходной БД.

Таблиця-шаблон

№	Преобразуемая таблица	Атрибут	Конструкция ЯМД	Оператор Раздел Влад	Раздел Влад	Оператор экзовлад	Экзовлад
1	2	3	4	5	6	7	8
1	Classifikator teknik	Код типа техники	<ЗначХО>=	<Раздел>=	X_УГЭС	<ЭкзО>=	#Наименование типа техники
2	Classifikator teknik	Код подтипа техники	<ЗначХО>=	<Раздел>=	X_УГЭС	<ЭкзО>=	#Наименование типа техники
3	Classifikator teknik	Код наименования техники	<ЗначХО>=	<Раздел>=	X_УГЭС	<ЭкзО>=	#Наименование подтипа техники
4	Nal_HUST	Марка двигателя (M_DVIG)		<Раздел>=	X_УГЭС		
5	Vodit_UGES	Номер водительских прав	<ЗначХС>=	<Раздел>=	X_УГЭС	<ЭкзО>=	#ФИО водителя
6	Vodit_UGES	Категория прав	<ЗначХС>=	<Раздел>=	X_УГЭС	<ЭкзО>=	#ФИО водителя

Оператор классэкзовл	Классо экзовлад	Оператор типэкзовл	Типо-экзовл	Оператор 1	Значение 1	№
9	10	11	12	13	14	1
<КлассО>=	Классификация техники	<ТипО>=	-	<ТипХО>[] (строковая,любая, фактическая)=	Код типа техники	1
<КлассО>=	Классификация техники	<ТипО>=	-	<КлассО>=	Наим.подтипа техники	2
\$<КлассО>=классификация техники; <ТипО>=-; <ЭкзО>=#Наименование_типа_техники; / <КлассО>=	Наименование подтипа техники	<ТипО>=	-			3
<КлассО>=	Двигатель	<ТипО>=	#Марка двигателя (M_DVIG)			4
<КлассО>=	Водитель	<ТипО>=	-	\$<ВремяНС>=#Дата получения водительских прав;<ТипХС>[] (строковая,любая,1)=Номер водительских прав;<КлассС>=	Получение водительских прав	5
<КлассО>=	Водитель	<ТипО>=	-	\$<ВремяНС>=#Дата получения водительских прав;<ТипХС>[] (строковая, списочная,1)= Категория водительских прав;<КлассС>=	Получение водительских прав	6

5. *Просмотр результатов реинжиниринга.* С помощью специальной программы просмотра, проверяется правильность преобразования БД. В случае обнаружения несоответствий между описанием предметной области и занесенными в БД данными, последние с помощью различных приложений программного инструментария разработчика БД с УМД удаляются. После чего вносятся требуемые изменения в помещаемые в таблицу-шаблон строки метаописаний ЯМД, и процесс преобразования (п. 3...5) повторяется. Такое повторение осуществляется до тех пор, пока не будет найдено адекватное отображение ПрО в схеме БД с УМД.

Данный метод был апробирован на нескольких базах данных различных информационных систем, подвергнутых реинжинирингу [5]. В результате такого апробирования было выявлено, что, действуя последовательно в соответствии с этапами данного метода, можно достаточно быстро проводить реинжиниринг различной сложности баз данных для любой информационной системы и предметной области.

Выводы

Предлагаемый метод решения проблемы реинжиниринга баз данных, в отличие от существующих

методов, интегрирован на уровне методологии, основывающейся на универсальной модели данных. Он содержит комплексное, целостное решение задач реинжиниринга – улучшение характеристик качества базы данных, расширение ее функциональности, понижение стоимости ее сопровождения, вероятности возникновения значимых для заказчика рисков, уменьшение сроков работ по ее сопровождению. Благодаря нему стало проще использовать данные «устаревших» информационных систем в системе, основанной на новой технологии с последующим возможным упрощением комплексирования новых программных систем на основе использования готовых компонентов со стандартными интерфейсами.

Список литературы

1. Кузнецов С. Модель зрелости для управления данными [Электронный ресурс] / С. Кузнецов. – Режим доступа к ресурсу: <http://www.citforum.ru/computer/2007-04/>.
2. Ахтырченко К.В. Методы и технологии реинжиниринга ИС / К.В. Ахтырченко, Т.П. Сорокваша // Труды Института системного программирования РАН. – М.: ИСП РАН, 2003. – Т. 4. – С. 141-162.
3. Есин В.И. Универсальная модель данных и ее математические основы / В.И. Есин // Системи обробки інформації. – Х.: Харківський університет Повітряних Сил, 2011. – № 2(92). – С. 21-24.

4. Есин В.И. Язык для универсальной модели данных / В.И. Есин, М.В. Есина // Системы обработки информации. – Х.: Харківський університет Повітряних Сил, 2011. – № 5(95). – С. 193-197.

5. Сорока Л.С. Реинжиниринг существующих баз данных в базу с универсальной моделью данных и возможностью распараллеливания процессов / Л.С. Сорока, В.И. Есин, М.В. Есина // Сборник научных трудов Параллельная компьютерная алгебра: Всероссийская научная конференция с элементами научной школы для молодежи,

Ставрополь, 11-15 октября 2010 г. Ставропольский государственный университет. – Ставрополь: Издательско-информационный центр «Фабула», 2010. – С. 298-304.

Поступила в редколлегию 14.01.2012

Рецензент: д-р техн. наук, проф. Л.С. Сорока, Харьковский национальный университет им. В.Н. Каразина, Харьков.

РЕІНЖІНІРІНГ ІСНУЮЧИХ БАЗ ДАНИХ

В.І. Єсін

Описується проблема необхідності реінжинірингу існуючих баз даних інформаційних систем компанії. Вирішити дану проблему пропонується за допомогою методу, що спирається на використання бази даних з універсальною моделлю.

Ключові слова: реінжиніринг, універсальна модель даних, база даних, база даних з універсальною моделлю даних.

REENGINEERING OF EXISTING DATABASES

V.I. Yesin

The problem of necessity of reengineering of existing databases of information systems companies is described. Deciding this problem is the method, which is based on the use of database with a universal model.

Keywords: reengineering, universal data model, database, database with universal data model.