

УДК 004.65.3

Тимашов А. А., к.т.н.; **Росинский В. В.**, научн. сотр.
(Институт кибернетики им. В.М. Глушкова НАН Украины)

ОБ ОДНОМ ПОДХОДЕ АВТОМАТИЗИРОВАННОГО ЗАПОЛНЕНИЯ БАЗ ДАННЫХ КОРПОРАТИВНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ НА ОСНОВЕ ДОКУМЕНТОВ MICROSOFT OFFICE С ЛЕКСИЧЕСКИ НЕСОГЛАСОВАННЫМИ ДАННЫМИ

Тимашов О. О., Росинський В. В. Про один підхід автоматизованого заповнення баз даних корпоративних інформаційних систем на основі документів Microsoft Office з лексично неузгодженими даними. Розглядаються переваги застосування інструменту Microsoft SQL Server 2000 Data Transformation Services при організації імпорту / експорту даних в корпоративних інформаційних системах. Пропонується алгоритм автоматизованого заповнення баз даних на основі лексично неузгоджених даних. Демонструється процес створення пакета Data Transformation Services з використанням вищевказаного алгоритму реалізованого у вигляді додатку Win 32.

Ключові слова: КОРПОРАТИВНА ІНФОРМАЦІЙНА СИСТЕМА, БАЗА ДАНИХ MICROSOFT EXCEL, MICROSOFT SQL SERVER, DATA TRANSFORMATION SERVICES

Тимашов А. А., Росинский В. В. Об одном подходе автоматизированного заполнения баз данных корпоративных информационных систем на основе документов Microsoft Office с лексически несогласованными данными. Рассматриваются преимущества применения инструмента Microsoft SQL Server 2000 Data Transformation Services при организации импорта/экспорта данных в корпоративных информационных системах. Предлагается алгоритм автоматизированного заполнения баз данных на основе лексически несогласованных данных. Демонстрируется процесс создания пакета Data Transformation Services с использованием вышеуказанного алгоритма реалізованого в виде приложения Win 32.

Ключевые слова: КОРПОРАТИВНАЯ ИНФОРМАЦИОННАЯ СИСТЕМА, БАЗА ДАННЫХ, MICROSOFT EXCEL, MICROSOFT SQL SERVER, DATA TRANSFORMATION SERVICES

Tymashov O. O., Rosynskiy V. V. About one approach of the automated filling of bases of corporate informative systems on the basis of documents of Microsoft Office with lexically uncoordinated information. Advantages of application of instrument of Microsoft SQL Server 2000 Data Transformation Services are examined during organization of import/export of information in the corporate informative systems. The algorithm of the automated filling of databases is offered on the basis of lexically uncoordinated information. The process of creation of package of Data Transformation Services is demonstrated with the use of foregoing algorithm realized as an Win 32 application.

Key words: CORPORATE INFORMATIVE SYSTEM, DATA BASE, MICROSOFT EXCEL, MICROSOFT SQL SERVER, DATA TRANSFORMATION SERVICES

Введение. По мере развития производства управленческий состав предприятий отмечает постоянное усложнение внутренних и внешних бизнес-процессов. Вновь сформировавшееся предприятие, которое не имеет больших оборотов, зачастую обходится стандартными программными продуктами MS Office, поскольку в таких условиях серьезная корпоративная информационная система (КИС) не нужна. MS Office позволяет на достаточно высоком уровне вести складской, бухгалтерский, налоговый и другие виды необходимых учетов. По мере расширения и увеличения деятельности предприятия растет и количество информации, которая поступает из внешних источников. Для принятия обоснованного решения такая информация должна быть достоверной, полной, непротиворечивой и адекватной. Именно на этом этапе возникает необходимость во внедрении КИС, которая представляет собой универсальный инструмент ведения хозяйственной деятельности предприятия. С ее помощью можно повысить эффективность операций во всех сферах деятельности предприятия, улучшая, таким образом, его экономические параметры. Основная задача КИС ускорить протекания основополагающих процессов предприятия и предоставить актуальную информацию.

В процессе создания КИС одной из главных проблем, которая возникает, является обмен данными между различными подсистемами, в том числе и унаследованными. Среди большинства специалистов бытует мнение, что решение задач обмена данными КИС с внешними системами лежат в физической плоскости и решены априори, поэтому необходимого внимания этим вопросам при анализе не уделяется. И как следствие вся нагрузка по созданию систем обмена данными ложится на проектировщиков.

Опыт организации обмена данными в КИС. Частные решения задачи обмена данных предлагают поставщики реляционных систем управления базами данных (СУБД) таких как IBM, Oracle, Informix, Sybase и т.д. Они поддерживают в своих продуктах шлюзы для доступа к базам данных, управляемых СУБД других поставщиков. Очень сильное воздействие на организацию корпоративных информационных систем с несколькими базами данных оказала разработка компанией Microsoft интерфейса ODBC (Open Database Connectivity) и появление соответствующей спецификации CLI (Call Level Interface) в стандарте языка SQL. Стандартизация библиотеки функций и подмножества языка SQL, обеспечивающих возможность обращаться к любой базе данных, для которой поддерживается соответствующий ODBC-драйвер, сильно снизило остроту проблемы интеграции данных. Практически все поставщики коммерческих и свободно доступных СУБД обеспечили ODBC-драйвера для доступа к своим базам данных, а открытость стандарта позволила создавать такие драйверы сторонним разработчикам. Однако ODBC и появившийся позже интерфейс JDBC (Java Database Connectivity) обеспечивает не интеграцию данных, а всего лишь доступ из приложения к различным базам данных, но, во-первых, часто этого оказывается достаточно, и, во-вторых, ограниченность технологии окупается ее простотой и эффективностью [1].

Задача обмена существенно облегчается, если данные определенного класса будут перемещаться между подсистемами, при условии, что в этих подсистемах будет заложена технологически реализованная возможность воспринимать извне и отдавать наружу данные в стандартном формате импорта/экспорта. Данный подход является основой для разработки метаданных и интерфейсов для обмена регулярными данными для различных унаследованных разноформатных систем [2].

Интерфейсы обмена данными КИС с внешними системами можно разбить на следующие категории: *одноразовый* импорт данных, унаследованных, как правило, из старой системы; *периодический* обмен данными между узлами информационной системы (внутренний обмен); *периодический* обмен данных с другими информационными системами (внешний обмен).

При анализе задач загрузки и выгрузки данных необходимо рассмотреть: *подсистемы*, которые нуждаются в интерфейсе выгрузки данных и требования к интерфейсу загрузки данных из внешней системы; *периодичность* обмена данными и объем передаваемых данных; *степень* синхронизации двух систем; *возможные* методы транспортировки данных.

Также следует: *согласовать* формат данных для обмена; *определить* зависимости загрузки и выгрузки, например порядок выполнения операций; *определить* мероприятия, которые необходимо выполнить при сбое во время загрузки и выгрузки данных; *сформулировать* правила определения ошибочных записей (при загрузке); *определить* правила регистрации операций передачи и приема данных; *определить* график передачи данных (в большинстве информационных систем эти операции выполняются в ночное время); *составить* график разработки и тестирования собственных утилит или скриптов обмена данными; *составить* график разовой загрузки данных, наследуемых из старой системы, и подготовить методику проверки корректности этой операции [3].

Постановка задачи. В сфере бизнеса для отслеживания и управления товарами очень часто прибегают к использованию электронных таблиц Excel. С его помощью можно легко подсчитать количество каждого наименования товара в остатке с помощью простой формулы. Также можно задать нужный формат программы для профессионального составления счет-фактуры и расчета размер налогов и других издержек. С другой стороны данные таблиц Excel зачастую используются КИС. Поэтому на сегодняшний день возникла необходимость в организации обмена данными между КИС и электронными таблицами Excel. Но за счет того, что данные в БД и электронных таблицах могут иметь разные формы лексического представления одного и того же объекта, возникает проблема несогласованности вновь вводимых данных с теми что уже имеются в системе. А поскольку

количество импортируемых записей в БД может измеряться сотнями тысяч, то проконтролировать все лексические несоответствия и ошибки не представляется возможным.

Цель данной работы – провести исследование механизмов импорта/экспорта данных платформы SQL Server 2000 и электронных таблиц Excel и описать процесс создания пакетов обмена данными для КИС, а также предложить алгоритм автоматизированного заполнения БД КИС на основе лексически несогласованных данных из внешних систем.

Алгоритм обработки лексически несогласованных данных в структуре пакета DTS.

В последнее время на предприятиях все больше используется централизованная модель хранения данных. Она предполагает сбор информации со всего предприятия на один или несколько серверов БД КИС, где она обрабатывается и хранится. Кроме того централизованное хранение информации позволяет создавать высокоэффективные системы поддержки принятия решений, которые становятся все более популярными в последнее время. Тем не менее, для хранения информации часто применяются самые разнообразные системы, нередко использующие разные технологии и поэтому имеющие несовместимые форматы хранения данных. А объединение всех существующих данных в один комплекс является довольно сложной задачей.

С помощью инструмента Microsoft SQL Server 2000 под названием Data Transformation Services (DTS) можно создавать мощные автоматизированные системы, которые будут осуществлять импорт/экспорт данных из множества источников, и выполнять их преобразование (трансформацию). Кроме того механизм DTS позволяет запускать пользовательские приложения, которые могут осуществлять обработку данных любой сложности [4].

Апробации результатов статьи проводилась на основе БД *InfoDB* КИС одной из ведущих фармацевтических компаний. В рамках исследования и механизмов импорта/экспорта данных платформы SQL Server 2000 и электронных таблиц Excel в статье продемонстрирован процесс создания в БД *InfoDB* пакета DTS с последующей автоматизированной обработкой и заполнением БД. Алгоритм обработки реализован как составная часть пакета DTS в виде утилиты *ClientRegistrar*.

Структура пользовательских таблиц БД *InfoDB* представлена на рис. 1:

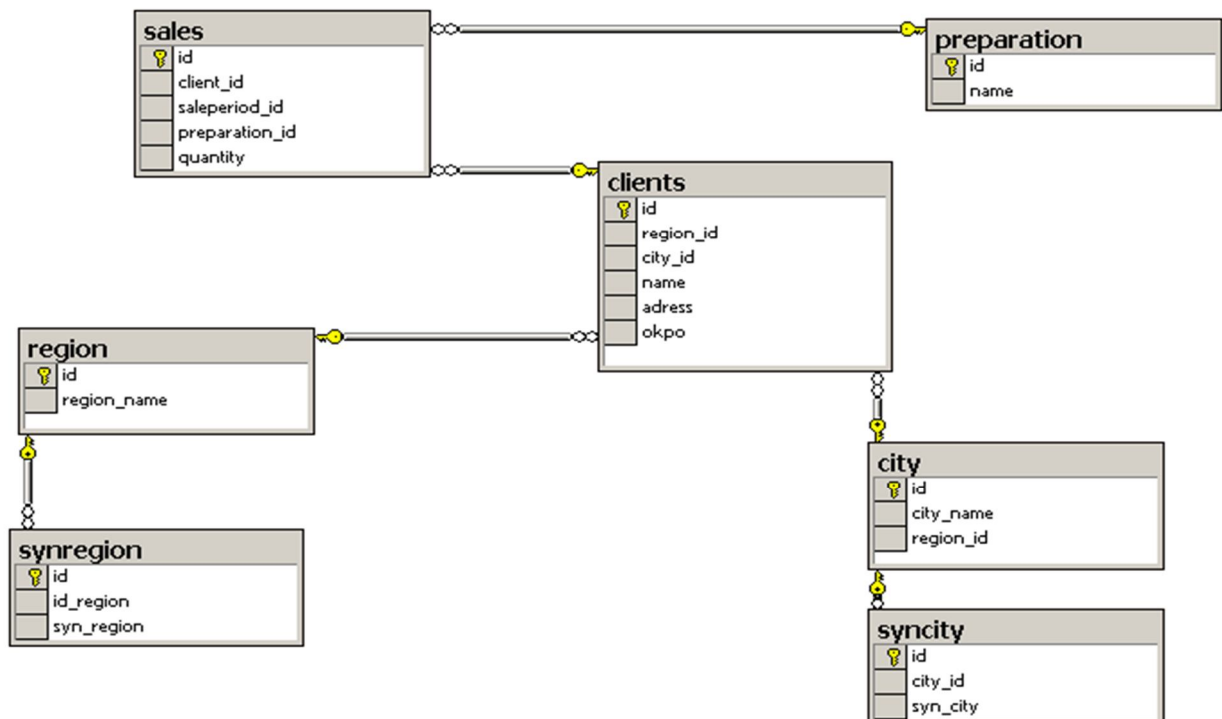


Рис. 1. Структура пользовательских таблиц БД InfoDB

Таблица *region* содержит названия всех областей, таблица *city* – названия всех городов. Таблица *clients* содержит информация по всем клиентам, которые когда-либо заказывали медицинские препараты компании. Названия самих препаратов хранятся в таблице *preparation*. Основной таблицей, которая предоставляет аналитическую информацию, является *sales*. На основе данных этой таблицы можно узнать, в какой отчетный период, каким клиентом, какой препарат и в каком количестве был закуплен. Кроме того имеются еще две таблицы *synregion* и *syncity* которые обеспечивают работу алгоритма автоматизированного заполнения БД КИС. В первой хранятся синонимы областей, которые попадают при копировании из документа Microsoft Excel в БД, а во второй – синонимы городов.

В качестве источника данных (рис. 2) для БД *InfoDB* выступает документ Microsoft Excel с именем *Sales.xls*, данные которого требуется скопировать в БД.

	A	B	C	D	E	F	G	H
1	Регион	Город	Физ.адрес	Покупатель	ОКПО	Товар	Кол-во	
2	ВИННИЦКАЯ	ВИННИЦА	м. Вінниця, вул. Івана	Єврофарм-Вінниця	30594322	Береш Магний	2	
3	ВИННИЦКАЯ ОБЛ.	ВИННИЦА	м. Вінниця, вул. Собор	Вінницяфармація	01974988	Активал Кид	6	
4	ВИННИЦКАЯ ОБЛ.	ВИННИЦА	м. Вінниця, вул. Собор	Вінницяфармація	01974988	Береш Магний	4	
5	ВИННИЦКАЯ	ВОРОНОВИЦА	смт.Вороновиця, вул. І	Богдан ПВКП, смт.	30257070	Береш Магний	2	
6	ВИННИЦКАЯ	ВОРОНОВИЦА	смт.Вороновиця, вул. І	Богдан ПВКП, смт.	30257070	Антифронт ка	3	
7	ВИННИЦКАЯ	м. ГАЙСИН	ул.К.Маркса,43	Строн ООО,Гайсын	20108923	Береш плюс н	3	
8	ВИННИЦКАЯ	м. ГАЙСИН	ул.К.Маркса,43	Строн ООО,Гайсын	20108923	Активал Кид	1	
9	ВИННИЦКАЯ	ГАЙСИН	м. Гайсин, вул. 1-го Тр	Поєдинчук В. М. ФС	2462609929	Береш плюс н	2	
10	ВИННИЦКАЯ	ГАЙСИН	м. Гайсин, вул. 1-го Тр	Поєдинчук В. М. ФС	2462609929	Береш Кальци	3	
11	ВИННИЦКАЯ ОБЛАСТЬ	ЖМЕРИНКА	м. Жмеринка, вул. І. Фі	Перша допомога ТС	34372340	Береш Магний	4	
12	ВИННИЦКАЯ ОБЛАСТЬ	ЖМЕРИНКА	м. Жмеринка, вул. І. Фі	Перша допомога ТС	34372340	Антифронт ка	2	
13	ВИННИЦКА ОБЛАСТЬ	м. ЖМЕРИНКА	ул. Пушкина, 1	ЦРА Жмеринская С	01975011	Береш Магний	2	

Рис. 2. Источник данных – документ Microsoft Excel Sales.xls

Документ представлен в виде таблицы, в кортежах которой отображают область, город, адрес, ОКПО и имя клиента, а также название купленного товара и его количество в отчетном периоде. Для переноса данных в БД сначала необходимо создать временную таблицу *sale* с полями, которые соответствуют полям таблицы документа *Sales.xls* и скопировать в нее содержимое документа. После чего следует запустить алгоритм автоматизированной обработки и заполнения БД. По завершению работы алгоритма, подготовленные данные следует добавить в таблицу БД *sales*.

В основе алгоритма автоматизированного заполнения БД КИС на основе лексически несогласованных данных лежит понятие логических дубликатов данных (ЛДД). ЛДД представляют собой единицу данных КИС, которые имеют одинаковую смысловую, но разную лексическую структуру. Чаще всего проблема ЛДД встречается при автоматизированном переносе данных из таблиц Microsoft Excel в БД КИС. Предлагаемый в статье метод решения проблемы ЛДД заключается в использовании таблиц синонимов. Таблицы синонимов создаются в БД и хранят синонимы данных, которые содержатся в основных таблицах. При копировании новых данных через таблицу синонимов проверяется наличие в них ЛДД. Если таковые имеются, то установить логическую связь с данными БД будет не сложно. В противном случае необходимо в автоматизированном режиме проверить наличие логической связи. Если она существует, то в таблицу синонимов добавляются новые синонимы, если нет – то данные добавляются в основные таблицы БД.

Работа утилиты (рис. 3) заключается в том, чтобы на основе данных в кортежах таблицы *sale* можно было однозначно идентифицировать находится ли клиент, который делал покупку в БД или нет. Если предположить что во всех документах Microsoft Excel с данными

о продажах, которые поступают в КИС, названия областей и городов семантически имеют один и тот же вид, то однозначно идентифицировать клиента можно по трем параметрах: область, город код ОКПО. Однако если код ОКПО вследствие того что состоит из чисел идентифицируется однозначно, то область и город могут представлять из себя ЛДД, в логическое соответствие которым необходимо поставить данные, используемые в БД КИС.

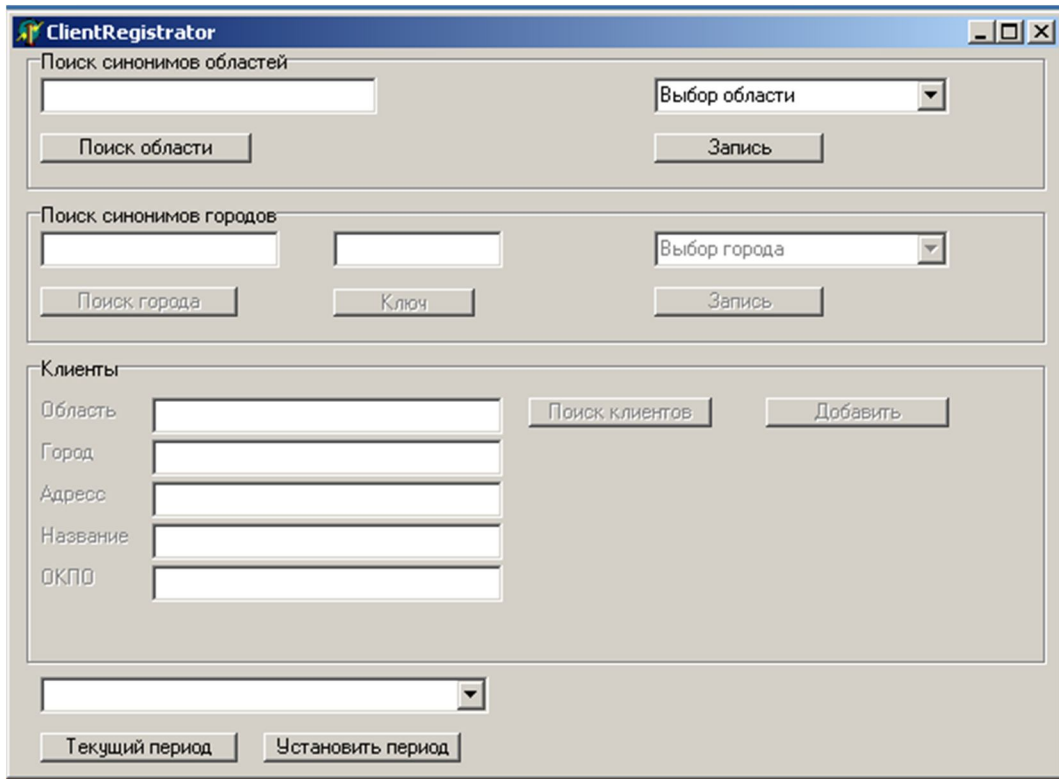


Рис. 3. Рабочее окно утилиты ClientRegistrar

При запуске утилиты доступна лишь вкладка для поиска ЛДД по измерению «область»(*region*), так как от этого измерения зависит измерение «город»(*city*). Если во временной таблице *sale*, которая заполняется на основе документа *Sales.xls* в поле *region* отсутствуют ЛДД из таблицы синонимов *synregion*, то при нажатии кнопки «Поиск области» один из них будет показан. В данном случае ЛДД представляет собой синоним одной из областей. После этого необходимо поставить ему в соответствие название области, по которому она идентифицируется в БД. Для этого в выпадающем списке «Выбор области» необходимо ее выбрать и нажать кнопку «Запись». После этого новый ЛДД попадет в таблицу синонимов *synregion*, и в дальнейшем ему можно будет поставить в соответствие данные из БД. Повторное нажатие кнопки «Поиск области» приведет к поиску нового ЛДД. Если таковых больше нет то вкладка «Поиск синонимов областей» станет недоступна, что будет означать что список ЛДД по измерению *region*, которые нельзя идентифицировать на основе данных таблицы *synregion* – исчерпан.

После этого, станет доступна вкладка «Поиск синонимов городов», которая была до этого не доступна. Поиск ЛДД по измерению *city* с помощью данной вкладки ничем не отличается от работы с вкладкой «Поиск синонимов областей», за исключением наличия поля «Ключ». Из множества кортежей временной таблице *sale*, выделяются те у которых можно идентифицировать все ЛДД по измерению *region*, а по измерению *city* – нельзя ни одного, поскольку эти ЛДД отсутствуют в таблице синонимов *syncity*. С учетом измерения *region* по нажатию кнопки «Поиск города» отображается ЛДД по измерению *city* одного из кортежей. В выпадающем списке «Выбор города» отобразится список всех городов области. При нажатии кнопки «Запись» новый ЛДД попадет в таблицу синонимов *syncity*. Для более быстрого поиска городов из выпадающего списка следует воспользоваться фильтром и

нажать кнопку «Ключ». После этого в выпадающем списке отобразятся лишь те города области, которые удовлетворяют условию фильтра. После того как все кортежи во временной таблице *sale* по измерению *city* можно будет идентифицировать на основе ЛДД из таблицы *syncity* вкладка «Поиск синонимов городов» станет недоступна.

На следующем шаге работы с утилитой *ClientRegistrar* станет доступной вкладка «Клиенты». На этом этапе работы алгоритма всем ЛДД в кортежах таблицы *sale* можно будет поставить в соответствие данных из БД и соответственно однозначно идентифицировать клиента. При нажатии кнопки «Поиск клиента» для правки будут предоставляться данные о клиентах, которых еще нет в БД. При нажатии кнопки «Добавить» откорректированные данные будут заноситься в таблицу *clients*. В самом низу окна утилиты можно определить к какому отчетному периоду будут относиться копируемые в БД данные и при необходимости изменить его.

Создать пакет DTS, использующий возможности служб трансформации данных в полной мере можно с помощью инструмента DTS Designer СУБД SQL Server 2000. Для запуска DTS Designer необходимо в панели Enterprise Manager выбрать нужный сервер и в контекстном меню папки **Data Transformation Services** выбрать команду **New Package**. В ответ откроется окно **DTS Package: <New Package>**.

Первая панель инструментов окна DTS Designer называется **Task** и содержит список *задач (task)*, поддерживаемых **SQL Server 2000**. Вторая панель называется **Connection** и содержит список *источников (datasource)*, с которыми может работать DTS Designer. Остальную часть окна инструмента занимает рабочая область.

Конфигурирование источников данных при работе с пакетами DTS осуществляется с помощью объектов **Connection objects**. В DTS Designer имеется множество шаблонных объектов, которые предназначены для установления соединения с распространенными источниками данных.

Чтобы сконфигурировать источник данных в пакете DTS, создаваемом с помощью DTS Designer, нужно воспользоваться панелью инструментов **Connection**. После выбора источника данных появится окно его свойств, в котором потребуется указать специфические характеристики. В рамках решения задач статьи необходимо рассмотреть следующие объекты:

- Microsoft OLE DB Provider for SQL Server – используется для обращения к серверам Microsoft SQL Server;
- Microsoft Excel 97-2000 – позволяет обращаться к файлам электронных таблиц Microsoft Excel версий 97-2000.

Поскольку данные предназначенные для копирования в БД изначально находятся в документе Microsoft Excel, то для работы с ними в рабочую область DTS Designer с панели инструментов **Connection** необходимо добавить объект **Microsoft Excel 97-2000**. После нажатия на его значок на панели инструментов **Connection** откроется окно конфигурации свойств данного объекта. Переключатель следует оставить в положении **New Connection**. В поле напротив необходимо задать имя соединения – *Microsoft Excel 97-2000*. В выпадающем списке **Data Source** следует оставить *Microsoft Excel 97-2000*. В поле **File name** необходимо ввести путь к документу Microsoft Excel и нажать кнопку **<OK>**.

Для копирования данных в БД сначала необходимо установить и настроить с ней соединение. За это отвечает объект **Microsoft OLE DB Provider for SQL Server**. Для его конфигурирования в окне **Connection Properties** переключатель **Existing connection** следует оставить без изменений и в поле напротив необходимо выбрать *Microsoft OLE DB Provider for SQL Server*. В выпадающем списке **Data source** следует оставить *Microsoft OLE DB Provider for SQL Server*. В списке серверов необходимо выбрать нужный сервер, а в поле **Database** – БД *InfoDB* и нажать кнопку **<OK>**.

Для создания временной таблицы *sale* необходимо воспользоваться задачей **Execute SQL Task**. Помимо этого, с помощью рассматриваемой задачи можно присвоить значения глобальным переменным выполняемого пакета. Сначала в рабочую область окна DTS

Designer необходимо добавить три задачи типа **Execute SQL Task**. Назначение первой задачи будет состоять в удалении таблицы *sale*. С помощью второй задачи таблица *sale* будет создаваться заново, а третья задача будет копировать в таблицу *sales* согласованные данные. При конфигурировании задач **Execute SQL Task** необходимо сконфигурировать следующие элементы управления:

- В поле **Description** следует указать имя задачи, которое должно быть уникальным в пределах пакета. Первая задача должна идентифицироваться в пакете как **DeleteTableSale**, вторая – **CreateTableSale**, третья – **InsertTableSales**.

- В раскрывающемся списке **Existing connection** следует выбрать имя одного из источников данных, сконфигурированных в пакете. Для всех трех задач источником будет выступать сама БД, то есть **Microsoft OLE DB Provider for SQL Server**. В контексте выбранного источника данных и будут выполняться указанные команды.

- С помощью поля **Command time-out** можно ограничить время выполнения запроса, указав в нем количество секунд, в течение которых будет ожидаться завершение выполнения запроса. Если за указанный период времени запрос не завершится естественным путем, то его выполнение прервется принудительно и будет считаться, что задача завершилась неудачно. При указании значения 0 (по умолчанию) время выполнения запроса на уровне задачи не ограничивается. Данное поле следует оставить без изменений.

- В поле **SQL statement** задачи **DeleteTableSale** указывается команда **Transact-SQL**, которая удаляет таблицу *sale* из БД. В этом же поле задачи **CreateTableSale** необходимо ввести команду **Transact-SQL**, которая создает таблицу *sale*. В поле **SQL statement** задачи **InsertTableSales** необходимо указать имя хранимой процедуры, которая вставляет согласованные данные в БД.

- Задача **Execute SQL Task** допускает использование параметризованных запросов, ссылающихся на глобальные переменные пакета. Поскольку параметры могут применяться не только при выполнении запросов, но также и в качестве аргументов хранимых процедур, то в качестве параметра хранимой процедуры необходимо ввести «?». Так как данные вносимые в БД отвечают определенному отчетному периоду, то в качестве параметра хранимой процедуры будет выступать идентификатор этого отчетного периода. Связывание параметра и глобальных переменных выполняется с помощью окна **Parameter Mapping**, которое открывается с помощью кнопки **Parameters** в окне свойств задачи. Окно имеет две вкладки – **Input Parameters** и **Output Parameters**. Первая из вкладок предназначена для связывания глобальных переменных с входными параметрами запроса, тогда как вторая — для присвоения результатов, возвращаемых запросом, глобальным переменным пакета. Прежде чем начать работу с входными параметрами необходимо создать глобальную переменную, в которой будет храниться значение отчетного периода. Для этого необходимо в рабочей области окна **DTS Designer** нажать правую кнопку мыши и в контекстном меню выбрать пункт **Package Properties**, а в нем перейти на вкладку **Global Variables**. В поле **Name** необходимо ввести имя переменной – **Saleperiod_id**. В поле **Type** необходимо указать ее тип – **Integer**. В поле **Value** задать ее начальное значение – 0. После этого нажать кнопку **<OK>**. Теперь на вкладке **Input Parameters** окна **Parameter Mapping** в поле **Input Global Variables** в выпадающем списке необходимо выбрать глобальную переменную **Saleperiod_id**. В поле **Parameters** значению глобальной переменной должен соответствовать параметр **Parameter1**.

Для дополнительной обработки данных на основе алгоритма синонимов необходимо воспользоваться задачей **Execute Process Task**. Эта задача в ходе выполнения пакета запускает приложения **Windows**. С помощью мыши необходимо добавить задачу в рабочую область окна **DTS Designer**. Конфигурирование задачи **Execute Process Task** выполняется следующим образом:

- В поле **Description** указывается имя, по которому будет выполняться идентификация задачи. Имя задачи должно быть уникально в пределах пакета.

- В поле Win32 process необходимо указать командную строку, которая будет выполняться каждый раз при запуске задачи. Фактически необходимо задать имя исполняемого файла (com, exe, bat), а также полный путь к нему. Эту информацию можно ввести либо вручную, либо выбрать нужный файл в окне Select File.

- Если приложение должно быть запущено со специфическими параметрами, то их указывают в поле Parameters.

- В поле Return code указывается значение, при возвращении которого приложением будет считаться, что оно выполнилось успешно. При возврате иного кода считается, что приложение завершилось с ошибкой и, как следствие, задача Execute Process Task считается выполненной неудачно.

- С помощью поля Timeout указывается количество секунд, в течение которого будет ожидаться завершение работы запущенного приложения. Если в указанный срок приложение не завершит работу, то будет считаться, что задача Execute Process Task выполнена неудачно. При указании значения 0 (по умолчанию) время ожидания завершения приложения не ограничивается.

- Флажок Terminate process after timeout становится доступным при указании в поле Timeout ненулевого значения. Если флажок установлен, то по истечении указанного периода ожидания система будет принудительно закрывать соответствующее приложение. Это необходимо, если пакет выполняется автоматически и требуется обеспечить продолжительную работу сервера.

Задача **Transform Data Task** выполняет копирования данных между документом Microsoft Excel и БД. Сначала в рабочей области пакета необходимо выбрать источник данных *Microsoft Excel*, который будет рассматриваться как экспортер, а затем, удерживая нажатой кнопку клавиатуры <Ctrl>, выбрать *Microsoft OLE DB Provider for SQL Server* – импортер данных. После выбора задачи **Transform Data Task** между выбранными источниками данных будет установлена связь. Окно свойств задачи **Transform Data Task** имеет пять вкладок.

Первая из вкладок называется **Source** и предназначена для определения источника, из которого будут получаться данные для копирования. В ней необходимо сконфигурировать следующие элементы управления:

- В поле **Description** необходимо указать имя, присваиваемое создаваемой задаче, которое должно быть уникально в пределах текущего пакета.

- В поле **Connection** отображается имя объекта источника данных, который был выбран в качестве экспортера данных при конфигурировании задачи **Transform Data Task**. Ниже на вкладке расположен переключатель, с помощью которого можно указать, будут ли данные браться из таблицы или получаться с помощью запроса.

- Переключатель следует оставить в положение **Table/View**, а в расположенном правее раскрывающемся списке необходимо задать имя листа документа Microsoft Excel. Все данные, содержащиеся в указанном листе, будут включены в процесс копирования. Нажав кнопку **Preview**, с помощью окна **View Data** можно просмотреть, какие данные содержатся в выбранной таблице или представлении подлежат копированию.

Вторая вкладка окна свойств задачи **Transform Data Task** называется **Destination** и предназначена для определения получателя данных. В поле **Connection** указывается имя источника данных, сконфигурированного в пакете и рассматриваемого в пределах текущей задачи в качестве импортера данных. Весь набор данных, подготовленный на вкладке **Source**, будет скопирован в указанную на вкладке **Destination** таблицу. В раскрывающемся списке **Table name**, который содержит имена всех таблиц, имеющихся на указанном источнике данных, необходимо выбрать имя таблицы *sale*.

Центральную часть вкладки занимает область, в которой перечисляются все столбцы, имеющиеся в таблице, выбранной в раскрывающемся списке **Table name**. Для каждого

столбца отображается его имя, тип данных, возможность хранения значений *NULL*, размер и точность (для нецелочисленных типов данных).

Третья вкладка окна свойств называется **Transformations** и предназначена для определения того, как должен происходить процесс копирования данных. Необходимо определить, какой столбец исходного набора данных, в какой столбец выбранной таблицы получателя будет копироваться. Кроме того, в процессе копирования можно использовать сценарии скриптовых языков ActiveX, что позволяет выполнять в ходе дублирования довольно сложные преобразования данных.

В верхней части вкладки имеется раскрывающийся список **Name**, в котором выбирается имя преобразования, с которым предполагается работать.

В списке **Source**, размещенном в левой части вкладки, указываются имена всех столбцов исходного набора данных, подготовленного с помощью вкладки **Source**. В данном случае именами столбцов будут имена столбцов документа Microsoft Excel *MonthSales*, данные из которого необходимо скопировать в БД *InfoDB*. Список **Destination** содержит перечень столбцов таблицы, выбранной на вкладке **Destination** в качестве получателя данных.

Связь между исходным столбцом и столбцом-получателем задается в виде соединительной линии (рис. 4). По умолчанию выполняется копирование всех столбцов исходного набора данных во все столбцы таблицы получателя, т. е. первый столбец исходных данных связывается с первым столбцом таблицы получателя, второй – со вторым и т. д. Если требуется изменить порядок, то необходимо перетащить конец стрелки на нужный столбец

По умолчанию для каждой пары связываемых столбцов создается отдельное преобразование, для которого указывается имя **DTSTransformation_N**, где вместо символа **N** подставляет номер преобразования. Преобразования при выполнении задачи производятся последовательно, а не одновременно. Для удаления имеющегося преобразования достаточно выбрать соответствующую связь и нажать кнопку **Delete**. Остальные вкладки окна свойств задачи Transform Data Task можно оставить без изменения.

Пакет DTS представляет собой довольно сложный объект, имеющий множество свойств. При конфигурировании пакета с помощью DTS Designer можно изменять любые свойства, как самого пакета, так и входящих в него соединений и задач. Однако иногда требуется изменить некоторые свойства в ходе выполнения пакета. В SQL Server 2000 управление свойствами пакета в ходе выполнения осуществляется с помощью задачи **Dynamic Properties Task**. С ее помощью на этапе выполнения пакета необходимо изменить значение глобальной переменной *Saleperiod_id*.

Окно свойств задачи состоит из одной вкладки. В верхней части окна расположено поле **Description** для указания имени, по которому конфигурируемая задача будет идентифицироваться в пакете. Это имя должно быть уникальным в пределах пакета. Основное пространство в окне занимает таблица **Change list**, содержащая все свойства,

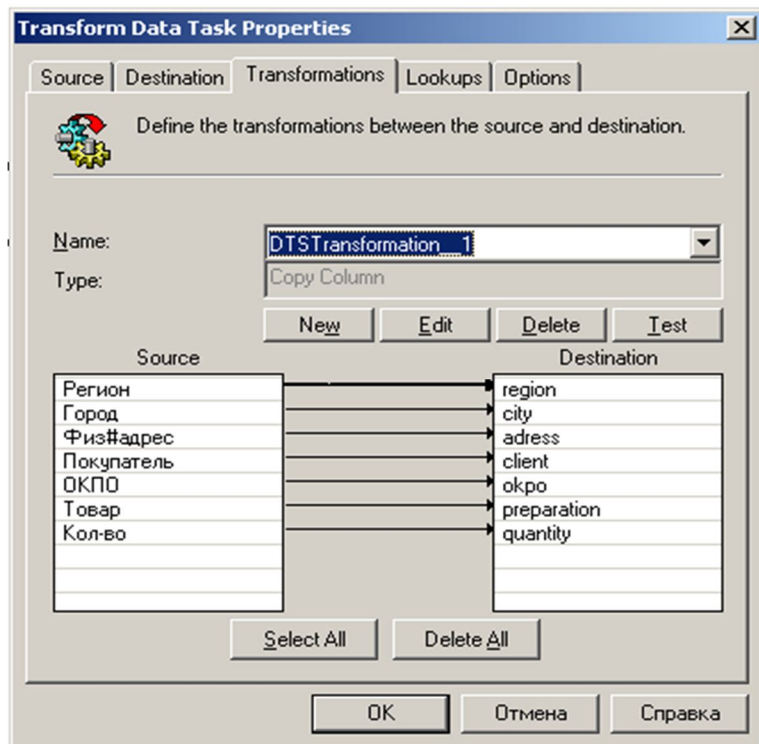


Рис. 4. Вкладка Transformations задачи Transform Data Task

которые будут изменяться конфигурируемой задачей. Для конфигурирования изменения свойств, необходимо нажать кнопку **Add**. После чего откроется окно **Dynamic Properties Task:Package Properties**, в котором перечислены все свойства, используемые в работе пакета. В списке **Global Variables** необходимо выбрать переменную *Saleperiod_id*. В правой части окна отображается набор свойств объекта, выбранного в левой части. Информация о свойствах выводится в виде таблицы, которая имеет колонки **Property Name** и **Default Value**. Значение **Default Value** следует установить равным 0

Для присвоения свойству нового значения необходимо выбрать его в правой части окна и нажать кнопку **Set**. В ответ откроется окно **Add/Edit Assignment**. С помощью этого окна необходимо указать значение, которое будет присвоено соответствующему свойству в ходе выполнения конфигурируемой задачи. С помощью списка **Source** необходимо выбрать способ получения значения, которое будет присвоено соответствующему свойству. В рамках данной задачи необходимо выбрать значение **Query**. В этом случае значение для присвоения получается в результате выполнения обычного запроса **Transact-SQL**. При конфигурировании запроса в первую очередь необходимо в раскрывающемся списке **Connection** выбрать имя источника данных *Microsoft OLE DB Provider for SQL Server*. В контексте этого источника и будет выполняться запрос. Код запроса указывается в поле **Query**. Результат запроса к таблице *curperiod* БД *InfoDB* возвращает значение отчетного периода, которое установлено посредством утилиты *ClientRegistrar* и записывает его в глобальную переменную *Saleperiod_id*.

Задачи, имеющиеся в пакете, могут выполняться либо независимо друг от друга, либо в определенной последовательности. В первом случае работа каждой из задач начинается в момент запуска пакета. Когда будет закончено выполнение последней задачи, работа всего пакета **DTS** будет завершена. Однако подобный подход целесообразно использовать лишь при работе с простыми пакетами, содержащими небольшое количество задач. При создании больших пакетов, включающих множество задач, требуется выполнять задачи в определенной последовательности. Это требование может быть реализовано с помощью отношений предшествования, определяющих порядок выполнения задач в пакете.

Отношение предшествования устанавливается между двумя задачами и бывает трех типов:

- **Безусловное (Unconditional)**. Означает безусловное выполнение одной задачи после завершения другой.
- **Успешное (On Success)**. При использовании отношения предшествования данного типа вторая задача будет выполнена только в том случае, когда первая задача успешно завершена.
- **Неудачное (On Failure)**. Это отношение предшествования используется, когда необходимо выполнить вторую задачу в случае неудачного завершения первой.

Для определения отношения предшествования необходимо с помощью клавиши **<Ctrl>** выбрать две задачи и указать в меню **Workflow** нужный тип отношения предшествования.

После этого задачи соединятся линией, причем со стороны подчиненной задачи линия будет иметь вид стрелки. В зависимости от используемого типа отношения предшествования меняется и цвет линии — **On Completion** отображается синим цветом, **On Success** — зеленым, **On Failure** — красным.

Первой в списке задач является **DeleteTableSale**, затем **CreateTableSale**. Между ними следует установить отношение *On Completion*. Между задачей **CreateTableSale** и объектом **Microsoft Excel 97-2000** следует установить отношение *On Success*, поскольку нет смысла начинать перенос данных из документа *Microsoft Excel MonthSales*, во временную таблицу *sale*, и соответственно, если она не будет создана продолжать работу пакета **DTS**. Объекты **Microsoft Excel 97-2000** и **Microsoft OLE DB Provider for SQL Server** уже связаны посредством задачи **Transform Data Task**. Данные скопированные из документа *Microsoft Excel MonthSales* во временную таблицу *sale* БД *InfoDB* посредством задачи **Transform Data Task** нуждаются в дополнительной обработке. Поскольку данная обработка должна быть произведена средствами утилиты *ClientRegistrar*, то следующей в списке задач должна

быть **Execute Process Task**. Между ней и объектом **Microsoft OLE DB Provider for SQL Server** следует установить отношение предшествования *On Completion*. Поскольку в процессе работы пакета DTS значение отчетного периода посредством утилиты *ClientRegistrar* было изменено, то следующей в списке задач необходимо поместить **Dynamic Properties Task** и установить между нею и **Execute Process Task** и отношение предшествования *On Completion*. На завершающем этапе, когда данные, которые необходимо скопировать в БД – согласованы, необходимо добавить список задач **InsertTableSales**. Поскольку если в процессе работы пакета DTS одна из задач, выполнение которых являются для алгоритма критическими, не будет выполнена, заполнение таблицы *sales* будет не полным или совсем не произойдет. Исходя из этого, между задачей **InsertTableSales** и **Dynamic Properties Task** следует поставить отношение предшествования *On Success*.

Выводы. В данной работе акцентировано внимание разработчиков КИС на основе централизованной модели хранения информации, на проблему несогласованности поступающих данных. В рамках работы проведены исследования механизмов импорта/экспорта данных платформы SQL Server 2000 и электронных таблиц Excel. Описан процесс создания пакетов DTS, который обеспечивает обмен данными между БД КИС и документом Microsoft Excel. Предложен алгоритм автоматизированного заполнения БД КИС на основе лексически несогласованных данных из внешних систем, реализованный в виде приложения Win 32, которое внедрено в ряде КИС отечественных фармацевтических компаний.

Литература

1. Старых В. А. Организация обмена данными в разнородных информационных системах в базисе технологий веб-сервисов / В.А. Старых, С.Б. Дунаев, С.Д. Коровкин // Сборник трудов «XI Всероссийская научно-методическая конференция «Телематика 2004»». – 2004, – Санкт-Петербург. – С. 29-35.
2. Мельников П.П. Обмен данными в разнородных бизнес-приложениях: учебное пособие / П. П. Мельников. – М.: Финакадемия, 2010. – 102 с.
3. <http://www.compress.ru/article.aspx?id=12527&part=index11ext1>
4. Мамаев Е.В. Microsoft SQL Server 2000 в подлиннике / Е.В. Мамаев. – Санкт-Петербург: БХВ-Петербург, 2005. – 987 с

УДК 681.325

Козленко М. І., к.т.н. (Приватний вищий навчальний заклад "Галицька академія")

ЗАВАДОСТІЙКІСТЬ ПЕРЕДАВАННЯ ТА ПРИЙМАННЯ ІНФОРМАЦІЇ НА ОСНОВІ ШИРОКОСМУГОВИХ СИГНАЛІВ З КЕРОВАНОЮ ЕНТРОПІЄЮ

Козленко М. І. Завадостійкість передавання та приймання інформації на основі широкосмугових сигналів з керованою ентропією. Проведено дослідження завадостійкості передавання дискретних інформаційних повідомлень за допомогою широкосмугових сигналів з керованою ентропією розподілу ймовірностей амплітуд. Отримано аналітичні вирази для обчислення завадостійкості. Проведено моделювання в обчислювальному експерименті, яке підтвердило коректність отриманих результатів.

Ключові слова: ШИРОКОСМУГОВИЙ СИГНАЛ, КЕРОВАНА ЕНТРОПІЯ, ЗАВАДОСТІЙКІСТЬ, ТЕЛЕКОМУНІКАЦІЙНА СИСТЕМА, КОМП'ЮТЕРНА СИСТЕМА

Козленко Н. И. Помехоустойчивость передачи и приема информации на основе широкополосных сигналов с управляемой энтропией. Проведено исследование помехоустойчивости передачи дискретных информационных сообщений с помощью широкополосных сигналов с переменной энтропией распределения