УДК 004.624
DOI 10.25205/1818-7900-2020-18-1-5-15

Моделирование обмена транзакционными данными в корпоративных системах

В. О. Демиш, **Б. Н. Пищик**

Новосибирский государственный университет
Новосибирск, *Россия*

Институт вычислительных технологий СО РАН
Новосибирск, *Россия*

Аннотация

Анализируются проблемы проектирования корпоративных систем для автоматизации предприятий с несколькими разделами и видами учета. Решается частная задача взаимодействия оперативного и бухгалтерского контуров учета на примере работы с давальческим сырьем. Обозначена необходимость создания модели взаимодействия блоков учета, обеспечивающих корректность работы с транзакциями уровня бизнес-логики. Предложена соответствующая модель, позволяющая спроектировать интеграционное решение на базе нескольких подсистем с описанием обмена транзакционными данными.

Ключевые слова

разработка корпоративных систем, интеграция данных, автоматизация предприятий

Для цитирования

Демиш В. О., *Пищик Б. Н.* Моделирование обмена транзакционными данными в корпоративных системах // Вестник НГУ. Серия: Информационные технологии. 2020. Т. 18, № 1. С. 5–15. DOI 10.25205/1818-7900-2020-18-1-5-15

Modeling of Transaction Data Exchange in Enterprise Systems

V. O. Demish, **B. N. Pischik**

Novosibirsk State University
Novosibirsk, *Russian Federation*

Institute of Computational Technologies SB RAS
Novosibirsk, *Russian Federation*

Abstract

This paper analyzes the problems of designing enterprise systems for solving problems with several sections and types of accounting. The particular problem of the interaction of the operational and accounting circuits is solved by the example of work with tolling raw materials.

The necessity of creating a model for the interaction of different subsystems that ensure the correctness of work with transactions at the level of business logic is indicated. An appropriate model is proposed that allows design integration solutions based on several subsystems with a description of the exchange of transactional data.

Keywords

enterprise systems design, data integration, enterprise automation

For citation

Demish V. O., Pischik B. N. Modeling of Transaction Data Exchange in Enterprise Systems. *Vestnik NSU. Series: Information Technologies*, 2020, vol. 18, no. 1, p. 5–15. (in Russ.) DOI 10.25205/1818-7900-2020-18-1-5-15

© В. О. Демиш, Б. Н. Пищик, 2020

Введение

Вне зависимости от направления деятельности (торговля, производство, оказание услуг и др.), организациям необходимо вести бухгалтерский и налоговый учет. Для отражения информации о хозяйственной деятельности экономического субъекта в Российской Федерации оба вида учета закреплены законодательно в Федеральном законе № 402-ФЗ и налоговом кодексе. Общие положения налогового учета изложены в статье 313 НК РФ.

С другой стороны, руководителям компаний-холдингов требуется вести корпоративный учет, рассматривая несколько юридических лиц как единое целое. При этом в любом случае требуется бухгалтерская и налоговая отчетность по каждому юридическому лицу. Так появляется еще один вид учета – управленческий [1].

Одна и та же операция при отражении в разных видах учета требует разной степени детализации. Например, фиксация данных о приобретенных товарах может быть отражена так:

- БУ (бухгалтерский учет): общее количество + общая сумма;
- НУ (налоговый учет): общая сумма + сумма НДС;
- УУ (управленческий учет): детальное количество, цены, размещение по складам.

Для обеспечения естественного требования единовременной фиксации (без дублирования ввода данных) каждой операции данные всех видов учета должны быть взаимосвязаны.

Работы в области экономических наук ориентированы на выстраивание наиболее эффективного учета на предприятиях. Это порождает различные требования к средствам автоматизации, которые должны быть предусмотрены разработчиками программного обеспечения.

В тексте работы будем использовать следующие термины предметной области.

Раздел учета – выделенная функциональная часть программы, которая предназначена для решения определенного фиксированного круга задач. Например: Продажи, Закупки, Производство, Кадры и т. п.

Вид учета (в литературе можно также встретить термин «контур учета») – замкнутая область данных, в которой отражаются операции предприятия. Например: оперативный, бухгалтерский, налоговый контуры.

Возможные сочетания разделов и видов учета на предприятиях:

Раздел учета / Вид учета	БУ	НУ	УУ
Планирование	–	–	+
CRM	–	–	+
Продажи	+	+	+
Закупки	–	–	++
Склад	+	–	++
Кадры	+	–	+
Заработная плата	+	+	+
Регламентированная отчетность	+	+	–
...	...	...	...

Состояние вида учета в сочетании с разделом учета в данной таблице обозначено различными знаками:

– не ведется;
\+ ведется с минимальной детализацией;
++ ведется с подробной детализацией.

Деление крайне условно и разным предприятиям будут соответствовать разные комбинации подобных вариантов.

В рамках автоматизации хозяйственной деятельности организации важно отметить, что каждая ячейка рассмотренной таблицы (назовем ее блоком учета) – это отдельная предмет-

ная область, которая пересекается с другими, но при этом может быть автоматизирована посредством отдельной информационной системы. В общем случае для полного покрытия всех проблем автоматизации потребуется интеграционное решение, в котором решены задачи всех видов и разделов учета.

В данной работе рассматриваются проблемы реализации бизнес-логики (в соответствии с терминологией М. Фаулера в [2]) в корпоративных системах автоматизации хозяйственной деятельности предприятий, основанных на интеграции подсистем, работающих с различными блоками учета.

Данные, с которыми осуществляется работа на уровне бизнес-логики, удобно разделить на четыре группы [3]:

- метаданные – описание структур данных;
- НСИ (нормативно-справочная информация) – справочники, классификаторы и др. (практически не изменяются во все время жизни информационной системы);
- мастер-данные – это условно-постоянные данные, определяющие состав автоматизируемой предметной области и основу для описания бизнес-логики приложения;
- транзакционные (или операционные) данные отображают факты выполнения бизнес-операций в системе (примеры: выполнение платежа, поступление товаров, начисление заработной платы).

Полнота интеграционного решения рассматриваемой проблемы заключается в обеспечении единого справочника НСИ и мастер-данных для всех информационных систем, а также в корректном отражении операционных данных в каждой из них.

В работе рассмотрено решение такой проблемы на примере, включающем несколько блоков учета.

Пример сквозного учета

Один из содержательных примеров взаимодействия нескольких блоков учета – переработка давальческого сырья.

Для построения примера достаточно рассмотреть две организации (А и Б), занимающиеся торговлей. При этом в организации А есть блок производства, а организация Б самостоятельного производства не имеет, но использует схему переработки сырья на стороне с помощью организации А.

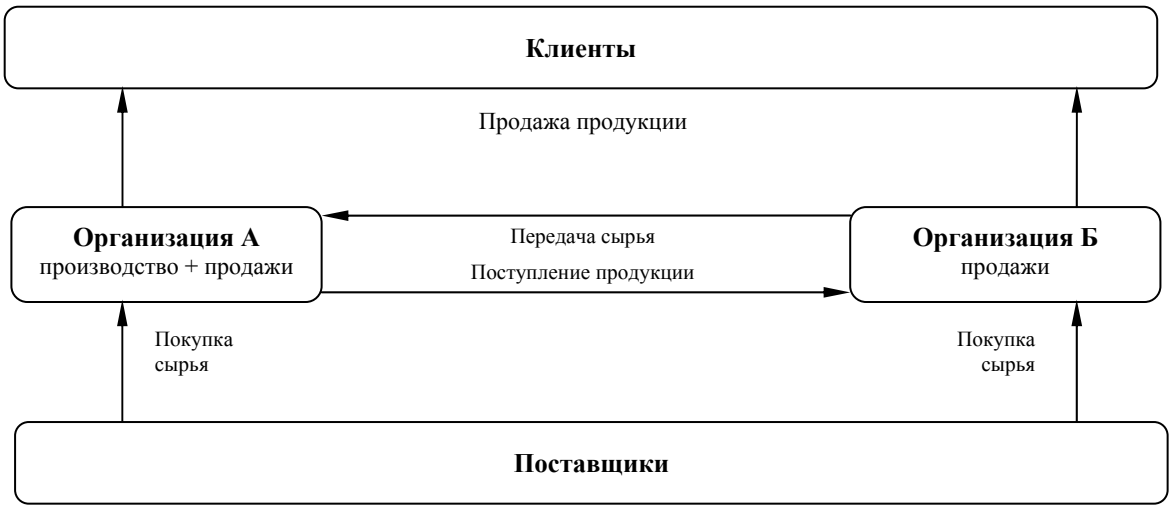

Рис. 1. Схема работы с давальческим сырьем
Fig. 1. Illustration of work with tolling raw materials

Такая схема работы – это один из видов *оптимизации налоговой нагрузки* [4], т. е. имеется экономически обоснованная необходимость автоматизации такой схемы работы. Пример показателен тем, что подсистемы бухгалтерского и управленческого учета рассматривают в нем бизнес-процессы по-разному. Для бухгалтерского учета важно четкое определение «хозяина» сырья и готовой продукции. В том случае, если организация А производит продукцию для организации Б, эта продукция будет учтена на так называемых забалансовых счетах и не будет числиться на балансе организации А. Но для руководителей отдела производства в данном случае важнее будет актуальный остаток произведенной продукции.

Также возможен сценарий, когда на этапе производства продукции из сырья еще не ясно, какой организацией она будет в дальнейшем реализована. Иначе говоря, организация А может произвести продукцию из собственного сырья, но в дальнейшем эту продукцию будет выгоднее реализовать от лица организации Б. И тогда нужно будет оформить производство в организации А через давальческую схему, а не из собственного сырья.

Какой вариант автоматизации такого бизнес-процесса будет наиболее оптимальным? Как предусмотреть возможность изменения данных «задним числом»? Как реализовать возможность однократного ввода данных лишь в одну из систем, если данные должны быть зафиксированы и в других системах?

Полные ответы требуют детального рассмотрения. В настоящей работе вместо этого предлагается провести обобщение примера до более широкого класса проблем и провести его анализ вне конкретной интерпретации.

Похоже, это один из тех примеров бизнес-логики, о которых упоминал М. Фаулер [2], называя их нелогичными по своей природе. Обилие таких примеров также присутствует в предметной области автоматизации учебных заведений [5].

Реализация примера

В рамках работы реализовано решение приведенного примера для конфигураций «ERP 2.4» [1] и «Бухгалтерия 3» [2] на платформе 1С: Предприятие. Выбранные инструменты актуальны для предприятий РФ ввиду распространенности продуктов 1С в сфере автоматизации предприятий.

Конфигурация ERP – это флагманское решение от 1С, позиционируемое как инструмент для автоматизации *всех контуров учета* предприятий.

В свою очередь, конфигурация «Бухгалтерия» автоматизирует только бухгалтерский учет. Это делает ее удобной для сдачи бухгалтерской отчетности, поскольку отражение всех операций на предприятии происходит с минимальной степенью детализации (необходимой для формирования отчетности).

Ключевая идея решения – использовать ERP в качестве основной информационной системы, но для ведения бухгалтерского учета использовать только конфигурацию «Бухгалтерия». В этом случае целесообразно внесение информации только в ERP с последующим обменом данными с системой для ведения бухгалтерии. Для этого реализованы следующие блоки:

- модуль переноса прочих операций, не относящихся к производству (закупки, продажи, кадры и расчет заработной платы);
- модуль итогового формирования объемов производства в организациях за заданный период;
- модуль переноса итоговых операций из ERP в Бухгалтерию.

[1] 1С: ERP Управление предприятием 2. URL: http://v8.1c.ru/erp/ (дата обращения 17.11.2019).
[2] "1С: Бухгалтерия 8» – доступно и всерьез. URL: http://v8.1c.ru/buhv8/ (дата обращения 17.11.2019).

Перенос прочих операций

Разработанный в 1С универсальный формат обмена данными в XML [3] позволяет переносить объекты данных 1С между различными конфигурациями. Но это решает задачу лишь частично, поскольку стандартные возможности формата не включают все блоки учета.

Еще один универсальный инструмент от 1С – конфигурация «Конвертация данных» [4] позволяет самостоятельно настраивать правила обмена между различными базами. С его помощью было настроено соответствие между объектами подсистемы расчета заработной платы в ERP и конфигурацией «Бухгалтерия».

Модуль итогового формирования объемов производства

Алгоритм формирования обобщенных итогов производства состоит из двух фаз. Первая – отбор реализованной продукции за период (месяц). Результатом этой фазы работы является таблица «Продажи за период» (Организация, Номенклатура, Количество, Продажи по давальческой схеме). Значение последней колонки истинно, если реализация была не от организации-производителя.

Вторая фаза – определение нужного объема сырья. Для этого требуется наличие действующих спецификаций производства на всю реализованную продукцию (количество сырья и других расходов на единицу продукции). Алгоритм должен учитывать многоэтапное производство. В таком случае на выпускаемые в процессе полуфабрикаты должна быть собственная спецификация. В результате работы второй фазы алгоритма на каждую пару (Организация, Продажи по давальческой схеме) формируется обобщенная таблица произведенной продукции и необходимого сырья для ее производства. Сохранив эту информацию в базе ERP, останется осуществить заключительный шаг – перенести ее в контур бухгалтерского учета.

Модуль переноса итоговых операций производства

Отображение информации по объему произведенной продукции и списанного сырья в блок бухгалтерского учета заключается в оформлении набора документов.

Для стороны давальца:

- передача на переработку;
- поступление из переработки.

Для переработчика:

- поступление сырья в переработку;
- передача давальческого сырья в производство;
- выпуск продукции из давальческого сырья;
- реализация услуг по переработке;
- передача продукции давальцу.

Вместе с этим для организации-производителя, которая выступает в качестве переработчика, также формируются документы для собственного производства:

- передача сырья в производство;
- выпуск собственной продукции.

[3] Универсальный механизм обмена данными. URL: http://v8.1c.ru/overview/Term_000000314.htm (дата обращения 17.11.2019).

[4] Организация конвертации данных и обмена данными с помощью конфигурации «Конвертация данных 2.0». URL: https://its.1c.ru/db/metod8dev/content/2943/hdoc (дата обращения 17.11.2019).

Для реализации такого отображения были созданы правила обмена данными в конфигурации «Конвертация данных». Получившееся решение представляет собой отображение подраздела производства из блока управленческого учета в контур бухгалтерского учета.

Периодичность обмена данными различна для разных компонентов. Основная часть данных (не относящаяся к работе с давальческим сырьем) выгружается из ERP в базу бухгалтерии по ходу добавления / изменения. А данные по давальческому сырью – только по итогам месяца.

Выбор платформы для решения задачи в пользу 1С позволяет снять ряд вопросов по соблюдению законодательных требований к бухгалтерскому учету, поскольку они решены в 1С, и больше внимания уделить проработке архитектуры предлагаемого решения.

Обобщение

Ориентируясь на приведенный пример, можно обобщить ситуации такого плана до более широкого класса задач, в которых несколько информационных систем реализуют определенный набор бизнес-операций и при этом осуществляют обмен следующими видами данных:

- НСИ;
- мастер-данные;
- транзакционные данные.

Данные могут различаться как по структуре, так и по степени детализации.

Отражение бизнес-операции в отдельных системах может включать неопределенность, которая разрешается спустя некоторое время. Так, объем продаж за месяц из приведенного примера становится полностью определенным только по истечении месяца. При этом в других системах этой неопределенности может не возникать ввиду различий отражения операций хозяйственной деятельности на уровне бизнес-логики.

Информация, необходимая для разрешения неопределенности в одной системе, может находиться в другой. Эта информация может быть представлена разными способами. Например, в системах, построенных на реляционных СУБД, это будут отношения и кортежи (таблицы и строки). Также это могут быть классы и объекты (если речь идет об объектно-ориентированном программировании). Можно воспользоваться подходом 1С и работать в контексте предметно-ориентированного программирования [6]. Такая парадигма позволяет оперировать предопределенными классами и объектами, характерными для сферы автоматизации учета на предприятиях. Например: Документ, Справочник, План счетов, Регистр бухгалтерии.

В рамках этой работы все эти варианты допустимы. Конечная реализация модели данных не столь важна – достаточно ограничиться описанием элементов данных как наборов атрибутов.

Моделирование решения

Транзакционные данные состоят из ссылок на НСИ и мастер-данные, а также свойств с атомарными значениями (числа, строки, даты). Например, данные о продаже продукции будут содержать ссылки на клиента, товар, дату операции, цену и др. Чтобы не было путаницы с понятием транзакций в СУБД, следует отметить, что все рассуждения касаются исключительно слоя бизнес-логики, но не слоя источников данных (где осуществляется взаимодействие с СУБД).

Фиксация транзакционных данных в систему может быть:

- *завершенная* – имеющая на момент фиксации все необходимые для отражения в учете данные;

- *частичная* – не имеющая на момент фиксации все необходимые для отражения в учете данные. Недостающие данные заполняются значениями по умолчанию либо остаются в исходном состоянии – не фиксируются.

Условие однократного ввода данных в одну из систем предполагает обязательный обмен данными между блоками учета. Таким образом, отображение блоков учета друг в (на) друга – это средство достижения согласованности данных в системе, если под согласованностью в данном случае понимать завершенность всех транзакций.

В связи с этим требуется разобраться в некоторых вопросах.

- Как определить, находится ли сейчас система в согласованном состоянии?
- Что требуется для достижения актуального состояния?
- Какими могут быть отображения блоков учета?

Рассмотрим построение модели распространения транзакции на примере двух информационных систем S и S'. Пусть все отражаемые в системах бизнес-операции (далее операции) делятся на три группы:

1) операции, отражаемые только в S;
2) операции, отражаемые только в S';
3) операции, изначально заносящие данные в S, но требующие отражения в том числе и в S'.

Все операции, относящиеся к первой и второй группам, завершаются сразу же после внесения их в S или S' соответственно. Третья группа вместе с отражением в S предполагает последующее отображение транзакционных данных в S'.

Для каждой операции A определен набор типов транзакций T_1, T_2, ..., T_n из S и T'_1, T'_2, … , T'_m из S', которые должны быть зафиксированы в каждой из систем. Для операций из первой группы $m = 0$, а из второй $n = 0$ соответственно.

Данные, которые фиксируются операцией A, в свою очередь, проецируются на НСИ и мастер-данные обеих систем. В рамках настоящей статьи предполагается, что синхронизация НСИ и мастер-данных осуществляется каким-то из возможных способов [3].

Рассмотрим, каким образом могут быть отображены транзакционные данные одной системы на данные другой системы.

Условимся, что все отображения делятся на инъекцию, сюръекцию и биекцию в соответствии с принятыми в математике определениями [7]. Если отображение является инъективным (в том числе биективным), то можно построить обратное отображение и, как следствие, использовать в качестве исходных данных любую из систем (а не только S). Иными словами, наличие инъективных отображений позволяет выбирать, в какую систему вносить данные в первую очередь, а в какую они должны отображаться. Это может быть полезным с практической точки зрения, например, исходя из удобства работы пользователей с какой-то конкретной системой.

Для решения рассмотренного примера «работа с давальческим сырьем» необходимы как минимум отображения вида

$$T_1, T_2, ..., T_n \to T'_1, T'_2, ..., T'_m, \tag{1}$$

$$\left(T_1, T_2, ..., T_n\right)\Delta t \to T'_1, T'_2, ..., T'_m, \tag{2}$$

где (1) включает преобразование транзакционных данных системы S в соответствующие данные системы S' посредством сопоставления их свойств и применения функциональных преобразований, доступных в системах.

Преобразования вида (2) более сложные и схожи с агрегатными функциями из языка SQL [8]. Они предполагают вычисление транзакций в S', основываясь на определенном временном промежутке Δt в системе S. Как вычисление итоговых сумм продаж за месяц в рассмотренном примере. При этом для разных операций соответствующий временной промежуток

может быть разным в зависимости от требований на уровне бизнес-логики (день, месяц, квартал и т. д.).

Несмотря на существенные отличия, преобразование вида (2) можно свести к (1), если ввести в системе S новый «обобщающий» тип транзакции, зависящий от других транзакций за определенный период времени:

$$T_{n+1} = \left(T_1, T_2, ..., T_n\right)\Delta t,$$

т. е. итоговая цепочка отображения данных из S в S' будет выглядеть так:

$$\left(T_1, T_2, ..., T_n\right)\Delta t \to T_{n+1} \to T_1', T_2', ..., T_m'.$$

Данные фиктивной транзакции T_{n+1} будут сохранены в S. В результате такого построения любая операция A может быть отображена в S' из S, когда данные за требуемый временной промежуток Δt будут внесены. До этих пор A будет находиться в очереди частично завершенных транзакций, ожидающих внесения недостающих данных в S. В зависимости от типа транзакционных данных можно разными способами реализовать выгрузку данных в S', например, по требованию пользователя (нажатием на кнопку), по заданному расписанию или с привязкой к заданным событиям в S.

Таким образом, *описание отображений транзакционных данных* между блоками учета позволяет определить корректность отражения операций предприятия в информационной системе.

Во-первых, дает возможность ответить на вопрос: «Можно ли единожды ввести операцию и впоследствии отразить ее на все блоки?».

Во-вторых, дает возможность иметь всю необходимую информацию о состоянии системы в каждый момент времени, а именно какие транзакционные данные уже отражены в системе в целом, какие находятся в стадии частичной фиксации, когда они будут отражены и что для этого необходимо.

Например, если операция A должна быть отражена в системе S транзакциями T_1, T_2, ..., T_n, T_{n+1}, T_{n+2}, ..., T_{n+k}, а в S' транзакциями T'_1, T'_2, ... , T'_m, T'_{m+1}, T'_{m+2}, ... , T'_{m+q}, , то отображение данных из S в S' может выглядеть как на рис. 2.

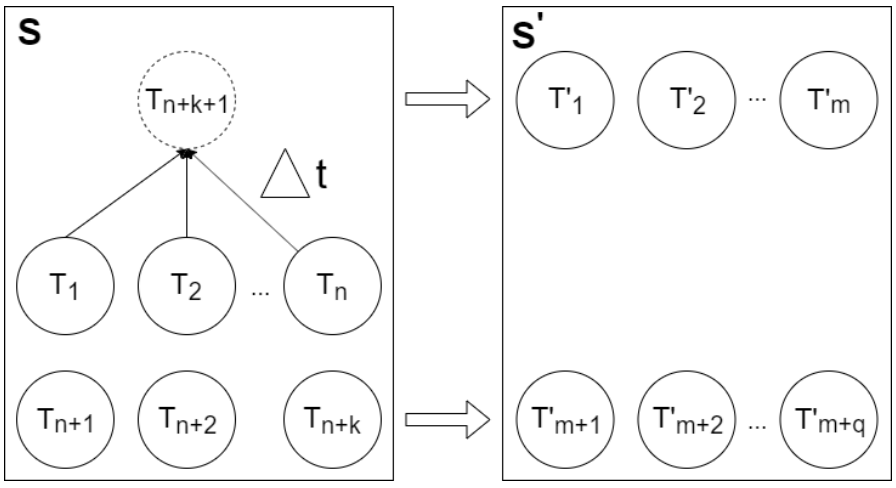

Рис. 2. Отображения транзакционных данных системы из S в S' на примере одной операции
Fig. 2. Example of transactional data export from S в S' for single operation

В этом случае часть транзакций в S' (это T'_1, T'_2, ... , T'_m) будут зафиксированы только по прошествии времени Δt , в то время как T'_{m+1}, T'_{m+2}, ... , T'_{m+q} сразу же после фиксации T_{n+1}, T_{n+2}, ... , T_{n+k} (посредством соответствующего отображения). Операция A не будет отражена полностью до тех пор, пока есть хотя бы одна не зафиксированная транзакция в какой-либо из систем. В свою очередь, общее интеграционное решение будет находиться в согласованном состоянии, только если все поступившие в него операции отражены полностью.

Реализация интеграционного решения на базе такой модели сводится к описанию обмена данными любым из возможных способов, рассматриваемых системах учета. В рассмотренном примере решения задачи на платформе 1С это был обмен xml-файлами по правилам, разработанным в традиционном для 1С инструментарии. Также это может быть использование общей БД (Shared DB), удаленный вызов процедур (Remote Procedure Call) или использование систем обмена сообщениями (Messaging) [9].

Заключение

В работе рассмотрены проблемы проектирования информационных систем для автоматизации хозяйственной деятельности предприятий. Ввиду необходимости вести несколько видов учета (бухгалтерский, налоговый, управленческий) предложено использование интеграционного решения на базе подсистем для каждого из блоков учета.

Построена модель описания отображений данных между информационными системами, позволяющая определить в любой момент времени степень завершенности транзакций бизнес-логики в системе. Для этого введена классификация состояний транзакций и видов отображений транзакционных данных между системами.

Модель была использована при решении задачи автоматизации работы нескольких организаций с давальческим сырьем для конфигураций «ERP 2.4» и «Бухгалтерия 3». Благодаря такому решению пользователям доступна информация о статусе всех операций, что позволяет не упустить из виду важные для отражения в учете данные. Ранее использовались эпизодические выгрузки данных, которые требовали ручного контроля со стороны пользователей. Несмотря на выбор платформы 1С в качестве решения приведенной задачи, предлагаемая модель также может использоваться для любой другой среды разработки.

Для полной модели более чем с двумя информационными системами отображение транзакционных данных значительно сложнее. Рассмотрение этого варианта в совокупности с разработкой инструмента для контроля завершенности транзакционных данных в целом – это предмет дальнейшего исследования.

Список литературы

1. **Друри К.** Управленческий и производственный учет. Вводный курс: Учебник. 5-е изд., перераб. и доп. М.: Юнити-Дана, 2014.
2. **Фаулер М.** Архитектура корпоративных программных приложений. М.: Вильямс, 2008. 544 с.
3. **Андрюшкевич С. К., Гуськов А. Е.** Практика решения задач интеграции информационных систем на основе управления мастер-данными // Вычислительные технологии. 2013. Т. 18, № 6. С. 3–15.
4. **Киркеева Л. И., Кушнер Е. Н.** Переработка давальческого сырья в АПК: бухгалтерский учет, налоговая оптимизация // Вестник Алтайского гос. аграр. ун-та. 2007. Вып. 2. С. 65–68.
5. **Демиш В. О., Пищик Б. Н., Козьменко Г. Г.** Проблемы автоматизации управления образовательным учреждением // Вестник НГУ. Серия: Информационные технологии. 2009. Т. 7, вып. 3. С. 94–102.

6. **Фаулер М.** Предметно-ориентированные языки программирования. М.: Вильямс, 2011. 576 с.
7. **Ершов Ю. Л., Палютин Е. А.** Математическая логика: Учеб. пособие. 3-е изд., стереотип. СПб.: Лань, 2004.
8. **Грабер М.** Введение в SQL. М.: Лори, 1992.
9. **Hohpe G., Woolf B.** Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. 1st ed. Addison-Wesley Professional, 2003.

References

1. **Drury C.** Management and production accounting. Introductory course. 5th ed. Moscow, Unity-Dana Publ., 2014. (in Russ.)
2. **Fowler M.** Architecture of corporate software applications. Moscow, Williams, 2008, 544 p. (in Russ.)
3. **Demish V. O., Pischik B. N., Kozmenko G. G.** Problems of automation of management of educational institution. *Vestnik NSU. Series: Information Technologies*, 2009, vol. 7, no. 3, p. 94–102. (in Russ.)
4. **Andryushkevich S. K., Guskov A. E.** Practice of solving problems of integration of information systems based on master data management. *Computing Technologies*, 2013, vol. 18, no. 6, p. 3–15. (in Russ.)
5. **Kirkeeva L. I., Kushner E. N.** Processing of tolling raw materials in agriculture: accounting, tax optimization. *Bulletin of the Altai State Agrarian University*, 2007, iss. 2, p. 65–68. (in Russ.)
6. **Fowler M.** Subject-oriented programming languages. Moscow, Williams Publ., 2011, 576 p. (in Russ.)
7. **Ershov Yu. L., Palyutin E. A.** Mathematical logic: A textbook. 3rd ed. Yeah, LAN, 2004, 336 p. (in Russ.)
8. **Graber M.** SQL. Moscow, Lori Publ., 1992. (in Russ.)

Материал поступил в редколлегию
Received
05.12.2019

Сведения об авторах

Демиш Всеволод Олегович, старший преподаватель, Новосибирский государственный университет (ул. Пирогова 2, Новосибирск, 630090, Россия); ведущий инженер-программист, Институт вычислительных технологий СО РАН (пр. Академика Лаврентьева, 6, Новосибирск, 630090, Россия)
v.demish@gmail.com

Пищик Борис Николаевич, кандидат технических наук, старший научный сотрудник, и.о. заведующего лабораторией, Институт вычислительных технологий СО РАН (пр. Академика Лаврентьева, 6, Новосибирск, 630090, Россия); доцент, заведующий кафедрой, Новосибирский государственный университет (ул. Пирогова 2, Новосибирск, 630090, Россия)
Boris.Pishchik@gmail.com

Information about the Authors

Vsevolod O. Demish, Lecturer, Novosibirsk State University (2 Pirogov Str., Novosibirsk, 630090, Russian Federation); Leading Software Engineer, Institute of Computational Technologies SB RAS (6 Academician Lavrentiev Ave., Novosibirsk, 630090, Russian Federation)
v.demish@gmail.com

Boris N. Pishchik, Cand. Tech. Sciences, Senior Researcher, Acting Head Laboratory, Institute of Computational Technologies SB RAS (6 Academician Lavrentiev Ave., Novosibirsk, 630090, Russian Federation); Associate Professor, Head of Department, Novosibirsk State University (2 Pirogov Str., Novosibirsk, 630090, Russian Federation)
Boris.Pishchik@gmail.com