

УДК 658.512: 004.62

К ВОПРОСУ ОБ АВТОМАТИЧЕСКОМ ПЕРЕНОСЕ ДАННЫХ ИЗ ОДНОЙ КОРПОРАТИВНОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ В ДРУГУЮ ПРИ РЕИНЖИНИРИНГЕ БИЗНЕС-ПРОЦЕССОВ ПРЕДПРИЯТИЯ

¹Дудолодов А.С., ²Гегечкори Е.Т.

¹ГК «Сатори-Партнер», Омск, e-mail: dudoladov@satoryomsk.ru;

²ФГБОУ ВО «Омский государственный технический университет», Омск, e-mail: getnbn@inbox.ru

В данной статье рассмотрена возможность интеграции и переноса данных между различными корпоративными информационными системами с отличающимися целями и способами ведения учета. В статье взяты за основу две КИС: 1С: Бухгалтерия Предприятия 3.0 ПРОФ и iiko. Соответственно требуется какая-либо программа/обработка, которая будет извлекать из одной КИС данные и создавать на их основе документы в другой КИС. Стоит отметить, что авторы лишь показывают, что какими бы разными системы не были, для чего бы не велся учет, всегда можно перенести данные, и самое главное, довести процесс переноса до автоматического уровня. Данная статья будет интересна программистам, бизнес-аналитикам и предпринимателям тех организаций, где учет ведется в нескольких программах.

Ключевые слова: корпоративная информационная система, учет, информационные технологии, автоматизация, реинжиниринг бизнес-процессов, бизнес-процессы

TO THE QUESTION OF AUTOMATIC TRANSFER OF DATA FROM ONE CORPORATE INFORMATION SYSTEM TO THE OTHER DURING ENTERPRISE BUSINESS PROCESS REENGINEERING

¹Dudoladov A.S., ²Gegechkori E.T.

¹Satori-Partner, Omsk, e-mail: dudoladov@satoryomsk.ru;

²Omsk State Technical University, Omsk, e-mail: getnbn@inbox.ru

This article discusses the possibility of integrating and transferring data between different corporate information systems with different purposes and methods of accounting. In this article, two CICs are used as a basis: 1C: Enterprise Accounting 3.0 PROF and iiko. Accordingly, a program is required that will extract data from one CIS and create documents based on them in another CIS. It is worth noting that in the article the authors only show that no matter what different systems are, why accounting is not maintained, it is always possible to transfer data and most importantly to automate the transfer process to the automatic level. This article will be interesting to programmers, business analysts and entrepreneurs, whose organizations are registered in several programs.

Keywords: corporate information system, accounting, information technology, automation, business process reengineering, business processes

На сегодняшний день каждое предприятие должно вести учет: оперативный, управленческий, бухгалтерский и другие. В отчетные периоды возникает необходимость сдавать отчетность в налоговую и другие контролирующие организации [1]. Отсюда и возникает вопрос: а что, если в организации управленческий учет ведется в одной информационной системе, а бухгалтерский учет в другой?

Для начала хотелось бы разобраться, почему может такое случиться. Во-первых: цель. Каждая корпоративная информационная система (далее – КИС) создавалась для решения определенных задач. То есть проблемы, решаемые в одной КИС, не могут быть решены в другой, если она для этого не предназначена. Во-вторых: целевая аудитория. Как известно, 1С: Бухгалтерия ориентируется на бухгалтерский учет, соответственно целевой аудиторией будут бухгалтеры и другие люди, которые как минимум знают основные бухгалтерские термины. Другие же КИС

ориентированы на людей, занимающихся закупками (storehouse), а третьи вообще на руководителей организации (iiko, «Большая птица» и другие).

Представим, что весь бухгалтерский учет на предприятии ведется в 1С: Бухгалтерии, а весь управленческий и оперативный – в iiko (организация занимается продажей пищевых продуктов в розницу). Когда возникнет необходимость сдавать отчетность, получится неоднозначная ситуация – данные в КИС не совпадают друг с другом. Что же делать в данной ситуации?

Можно вручную переносить данные, но это займет не один день (предположим, что организация имеет несколько филиалов в разных городах). Соответственно понадобится какая-либо программа/обработка, которая будет извлекать из одной КИС данные и создавать на их основе документы в другой КИС.

Вопросам, как это организовать, с чего начать и как в конечном итоге автоматизировать процесс, посвящена статья.

Для основы разработки взяты две КИС: 1С: Бухгалтерия и iiko. Данные будут переноситься из iiko в 1С на базе операционной системы Windows 7.

Стоит отметить, что, как правило, в пределах одной системы, но в разных базах существуют стандартные средства переноса данных. Делается это общей выгрузкой базы, сравнением и объединением, переносом документов и прочим. Также в системе 1С есть перенос данных как раз путем подключения к другой базе (перенос документов из 1С: Зарплата и управление персоналом в 1С: Бухгалтерия предприятия). Однако при таком переносе действуют определенные правила обмена, то есть невозможно перенести документ из одной базы в другую, если у документов в дереве конфигурации имеются разные реквизиты (система просто не сможет их между собой сопоставить). Поэтому потребуется написать свой конвертор, которому не будет важно, что мы перекидываем документы из разных КИС, так как внутри него будут прописаны правила сопоставления реквизитов.

Для начала нужно понимать, что есть возможность обратиться из одной базы к другой. Для этого нам понадобится функция с параметрами:

1. Путем расположения базы, из которой будут извлекаться данные.
2. Идентификационные данные подключения к базе.
3. Период, за который будет взята информация.
4. Тип загружаемых данных (лучше использовать тип – строка).

Схема необходимых данных, которые должен содержать в себе СОМ объект, должна быть строго задана и передаваться

через несколько параметров, таких как путь к базе и промежуток, за который нужна необходимая информация (рис. 1).

Данные действия необходимо сделать в любой системе, которая поддерживает создание dll файлов [2]. Этой системой вполне может выступить visual studio и язык С# или С++. Также вполне возможно использовать язык Java. После создания dll ее необходимо записать в реестр с помощью команды:

```
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\RegAsm.exe «путь к dll» /codebase /tlb: «Названиедll.tlb».
```

Для выполнения данной процедуры на компьютере обязательно должен быть установлен Microsoft.NET версии 4 или выше. Обязательно в параметрах записи указать codebase и tlb, в противном случае система не сможет вызвать функцию, содержащуюся в dll из базы, в которую будут загружаться документы [3].

Теперь у нас создан СОМ объект, содержащий акты реализации и отчеты о розничных продажах, перенесенных из iiko. Что же делать с этими данными? Ведь наименование контрагентов, номенклатуры, складов и организаций в разных КИС может не совпадать, так как данные, как правило, вводятся разными людьми.

Конечно же, мы можем вывести результат клиенту пользователю на форму, чтобы он вручную сопоставил данные и запустил обработку по проведению документов. Но чем же этот результат отличается от ручного заведения документов? Во-первых: все данные, перенесенные через СОМ объект, достоверные, пользователь не сможет ошибиться, а во-вторых, это значительно сократит время, затрачиваемое на перенос данных [4].

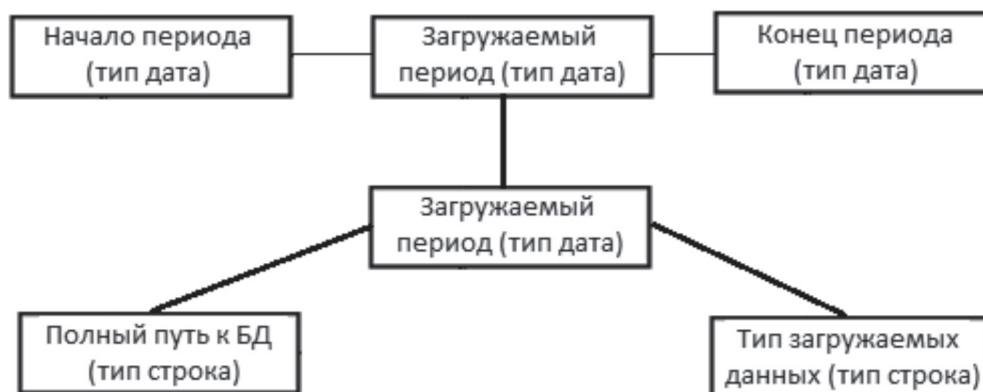


Рис. 1. Необходимые параметры для создания СОМ объекта

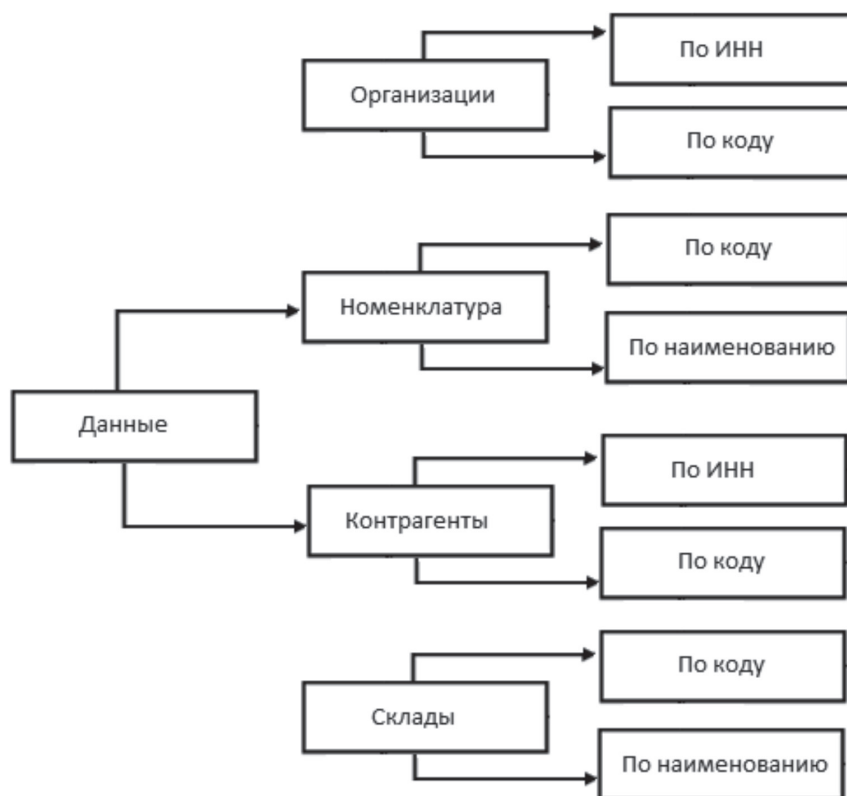


Рис. 2. Содержание автоматического сопоставления данных

То есть будет существовать клиентская форма, которая будет показывать все данные документов и справочников, перенесенные через СОМ объект из iiko, с одной стороны и пустыми полями со стороны 1С (эти поля будут иметь тип уже не строка, справочник/ документ ссылка). Можно будет выбрать из выпадающего списка контрагента, товар и другое или создать новый объект. После сопоставления данных обработка будет запускаться во второй раз, но так как все данные сопоставлены, уже будет производиться проведение документов.

На этом можно было и закончить, но проделывать сопоставление каждый раз вручную при сдаче отчетности проблематично. Поэтому можно шагнуть еще дальше и настроить автоматическое сопоставление.

В автоматическом сопоставлении можно указать перечень данных, которые будут сравниваться с данными, содержащимися в базе, в которую загружаются данные (рис. 2).

Если по какой-либо причине данные не были сопоставлены (ошибка на стороне базы, в которую загружаем, или отсутствие объекта в ней же), то нужно будет обра-

щаться к функции, которая будет создавать эти объекты.

К примеру, из iiko вытянулся документ, в котором в реквизите «контрагент» стоит клиент А. В 1С при сопоставлении контрагента А по ИНН и полю «код iiko» не нашлось совпадений. В таком случае обработка обращается к функции, создающей контрагента. ИНН, КПП, наименование, договор – все это уже есть в тех данных, которые обработка получила из iiko, так что создать объект контрагент А справочника «контрагенты» не составит никакого труда.

Такие функции должны быть созданы на все основные объекты конфигурации (организации, контрагенты, договоры, соглашения, номенклатура, склады и т.д.), и прописаны оператором условия в случае обращения обработки к ним.

Теперь при запуске обработки будет не только создаваться СОМ объект, вытягивающий данные из iiko, но и эти данные будут сопоставляться с данными из 1С для корректной загрузки документов.

Получается, что теперь достаточно перед сдачей отчетности запустить обработку

за определенный период и проводить документы [5]. Однако на этом этапе мы столкнемся с еще одной проблемой – временем. Создание СОМ объекта, дальнейшее сопоставление данных, проведение документов за 1 день может занимать от 5 минут до 2 часов в зависимости от: скорости соединения, технического оборудования, количества документов за данный день, количества работающих пользователей в КИС и множества других факторов [6].

Самым оптимальным решением будет перенос данных каждый день, по завершении работы. Однако экономически неэффективно держать человека, который будет раз в день запускать данную полуавтоматизированную обработку. Поэтому необходимо, чтобы данная обработка запускалась автоматически каждую ночь, после завершения работы всех пользователей. Однако, исходя из того, что было написано, часть кода обработки выполняется на сервере, так что, во-первых, понадобится перенести весь код в модуль объекта. После проведения данной процедуры весь код будет исполняться на сервере [7]. Но стоит отметить, что обращения к объектам базы с клиента и с сервера разные, так что значительную часть кода нужно будет переписать. Также нужно внимательно проверить сам код на доступность функций и процедур. К примеру, функция нажатия кнопки будет недоступна на сервере, как и с другой стороны запрос в базу данных недоступен на клиенте.

Но не стоит убирать клиентскую часть обработки, так как стоит понимать, что обработка работает, когда есть подключение к базе iiko. Его может не быть элементарно из-за поломки внутренней сети предприятия, доступности сервера, прав и прочего. В таком случае можно настроить отложен-

ный запуск. К примеру, если обработка не сработала, то повторить попытку запуска через час, после неудачного запуска. Обычно это работает, но в силу технических причин может не отработать.

В таком случае, нужно сделать протоколы загрузки документов (логи работы программы). В случаях, если обработка не сработала или сработала некорректно (не произошло проведение документов) должен сохраняться лог выгрузки, в котором будут подробно расписаны все ошибки программы, недоступность сервера и прочее. При необходимости можно проверить лог и запустить обработку вручную за невыгруженный период, ведь клиентская часть обработки не была полностью стерта.

После того, как код переписан, во избежание ошибок необходимо проверить его через пустую форму с кнопкой, которая будет вызывать автоматический запуск обработки. Если алгоритм сработал правильно, то в БД появятся документы, загружаемые из другой базы, за необходимый период.

Теперь можно настроить автоматический запуск обработки.

Для этого понадобится просто присвоить получившейся функции определенное время срабатывания и автоматический расчет необходимых параметров, так как невозможно будет их передать без кода на клиенте (рис. 1).

Также стоит отметить, что необходимо проверить права. Для регламентного задания, исполняющегося в пределах загружаемой базы, должны быть установлены полные права на создание СОМ объекта и подключение к базе. Если эти условия не будут соблюдены, тогда фоновое задание просто не сможет подключиться к функции dll [8].

Полный алгоритм предоставлен на рис. 3.

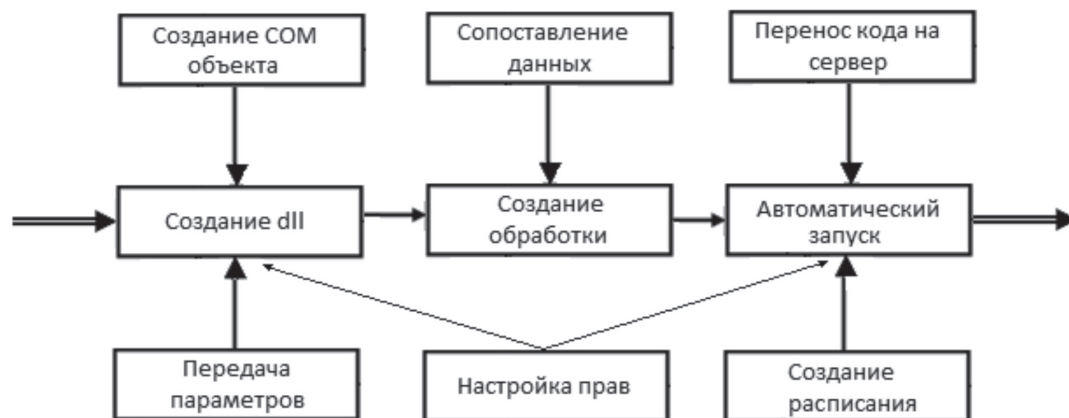


Рис. 3. Полный алгоритм переноса данных

Вывод: перенос данных из одной КИС в другую не только реален, но и достижим для исполнения на полном автомате. Стоит один раз автоматизировать этот процесс, и организация будет иметь:

1. Всегда достоверные и актуальные данные в каждой КИС в пределах одного дня.
2. Отсутствие проблем с отчетностью в контролирующие органы.
3. Отсутствие проблем внутреннего учета продукции.
4. Отсутствие контроля выполнения.

Данный алгоритм может работать не только с системами 1С и iiko, но и с другими, поменяется лишь язык кодирования, но смысл останется. Также алгоритм может отрабатывать и в другую сторону.

Примерное время разработки данного алгоритма и его реализация с тестированием заняли около 80 часов:

1. Разработка бизнес-модели обработки, сбор необходимого перечня документации, сбор начальных данных (5 часов).
2. Создание файла dll с правилами сбора данных из базы iiko (30 часов).
3. Создание обработки 1С, обращающейся к библиотеке dll и создающей СОМ объект, систематизирующей данные (20 часов).
4. Создание алгоритма сопоставления информации разных КИС, а также создание функций создания новых объектов конфигурации (10 часов).

5. Создание механизма автоматического запуска обработки и правила ведения логов некорректной выгрузки (10 часов).

6. Тестирование обработки (5 часов).

Хотелось бы отметить, что проделать описанную схему может как программист, работающий непосредственно на предприятии, так и заказной сторонний программист. Это, несомненно, потребует финансовых затрат, но механизм полностью себя оправдывает и окупится при самых неблагоприятных условиях за полгода. При благоприятных условиях – за 2–3 месяца.

Список литературы

1. Широков Л.А. Бухгалтерские информационные системы / Л.А. Широков. – М.: МГИУ, 2009. – 248 с.
2. Патрушина С.М. Информационные системы в бухгалтерском учете / С.М. Патрушина. – М.: МарТ, 2009. – 368 с.
3. Лойко В.И. Информационные системы и технологии в экономике / В.И. Лойко. – М.: Финансы и статистика, 2009. – 416 с.
4. Федотова Е.Л. Информационные технологии в профессиональной деятельности / Е.Л. Федотова. – М.: ИНФРА-М, 2008. – 145 с.
5. Национальные стандарты. Единая система программной документации. Издание официальное / М.: Стандартинформ, 2005. – 128 с.
6. Государственные стандарты. Информационная технология. Автоматизированные системы. Основные положения. Издание официальное. – М.: ИПК Издательство стандартов, 2002. – 176 с.
7. Жукова С.А. Моделирование и оптимизация структуры информационного ресурса: Автореф. дисс. канд. техн. наук. / С.А. Жукова. – Ижевск : ИжГТУ, 2008. – 18с.
8. Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем / А.М. Вендров. – М.: Финансы и статистика, 1998. – 176 с.