

## **ДРАКОН ПОМОГАЕТ БУХГАЛТЕРУ**

*Александр Шилин*

В современные системы бухгалтерского учета закладывается все больше сложных правил анализа и обработки информации. Эти правила очень трудно объяснить бухгалтеру, потому что они представляют собой длинные цепочки анализа и принятия решения. При внедрении бухгалтерских программ очень часто возникает проблема непонимания. Если бухгалтер не понимает, как программа формирует проводки, у него возникает недоверие. Недоверие порождает неумение и нежелание использовать программу.

Наша фирма «Инженеры информации» занимается внедрением программы 1С-Бухгалтерия 7.7. Программа поставляется в типовой конфигурации, которая содержит наиболее распространенные в бухгалтерском учете документы. Эти документы при заполнении формируют по заданным правилам бухгалтерские проводки. Проводки формируются в разных вариантах. Результат проводки зависит от остатков и оборотов бухгалтерских счетов в разные моменты времени. Трудность в том, что схема учета, принятая на предприятии, часто отличается от схемы учета заложенной в стандартной конфигурации.

Задача нашей фирмы — объяснить бухгалтеру схему учета, принятую в типовой конфигурации программы «1С-Бухгалтерия 7.7». Бухгалтер должен увидеть отличия своей схемы учета от типовой конфигурации. И принять решение о доработке типовой конфигурации.

### **Как мы использовали язык Дракон**

Мы долго искали метод доступного и наглядного изображения правил, заложенных в программу бухгалтерского учета. Год назад нам встретилась книга «В.Д. Паронджанов. как улучшить работу ума. Новые средства для образного представления знаний, развития интеллекта и взаимопонимания. М.: Радио и связь, 1998. 352с.». В ней описан язык визуального отображения информации Дракон. Он использовался при создании вычислительной системы космического корабля «Буран» и сейчас применяется в проектировании ракетной техники.

Методы и описания языка Дракон напоминают блок-схемы. Но в нем есть то, что отсутствует в блок-схемах. В язык Дракон заложена система правил, которые улучшают восприятие информации. Оформление блок-схем по этим правилам делает их понятным любому человеку.

Раньше (до знакомства с Драконом) мы были вынуждены описывать работу документов в текстовом виде с использованием текстовых алгоритмических конструкций. Приведем пример.

### **Пример текстовой записи работы документа Приходный Кассовый Ордер**

Работа с документом «Приходный Кассовый Ордер»  
Какой счет записан в поле Корреспондирующий счет  
Если счет 46 (Реализация) Тогда  
    Если в документе указан Налог С Продаж Тогда  
        Проводка Дебет(46) Кредит(68, 30)  
    Конец Если  
        Проводка Дебет(50) Кредит(46)  
Если счет 62 (Покупатели) Тогда  
    Обработка счета 62

Если счет 64 (Предоплата) Тогда  
 Если в документе указан Налог С Продаж Тогда  
 Если местные законы требуют платить с авансов Налог С Продаж Тогда  
 Начисляем Налог С Продаж с аванса  
 Проводка Дебет (64) Кредит (68, 30)  
 Конец Если  
 Конец Если  
 Выделяем НДС с аванса  
 Проводка Дебет (64) Кредит (68, 2)  
 Проводка Дебет (50) Кредит (64)  
 Если другой счет (не 46, 62, 64) Тогда  
 Проводка Дебет (50) Кредит (Корреспондирующий счет)  
 Конец Если  
 Конец работы с документом

## **Дракон-схема, описывающая работу документа Приходный Кассовый Ордер**

Пример дракон-схемы показан на рис. 141. Легко видеть, что содержание текста и схемы полностью совпадает. Основные предложения в тексте и надписи на рисунке не отличаются. Исчезли лишь ключевые слова: Если, Тогда, Конец Если и т.д. Вместо них появились линии, отражающие ход управления программой.

Сравнивая две формы представления алгоритма, легко убедиться, что дракон-схема намного удобнее. Благодаря схеме изложение мысли приобрело наглядность и доходчивость, свойственные графике.

Отметим важную особенность. В данном документе обработка счета 62 слишком объемна. Если показать ее в тексте или на схеме полностью, из-за большого объема будет потеряна ясность и наглядность. Поэтому обработка счета 62 выделена в отдельную процедуру.

В тексте она указана одной фразой «Обработка счета 62». В текстовой форме процедура приведена ниже. А дракон-схема показана на рис. 142.

## **Пример текстовой записи обработки счета 62 при поступлении денег в кассу**

Обработка счета 62  
 Проверить задолженность Покупателя  
 Если Есть задолженность у Покупателя Тогда  
 Если Покупатель платит больше чем должен нам Тогда  
 Предупредить бухгалтера об этом  
 Конец Если  
 Если Метод определения выручки По Оплате тогда  
 Если есть неоплаченная отгрузка Тогда  
 Вычислить какая часть долга Покупателя гасится суммой Оплаты  
 Пропорционально уменьшить сумму неоплаченной отгрузки  
 Проводка Дебет (ПС) (ПС – забалансовый счет) .  
 Конец Если  
 Конец Если  
 Конец Если  
 Если в документе указан Налог С Продаж Тогда  
 Начисляем неоплаченный при отгрузке Налог С Продаж.  
 Проводка Дебет (76, 4) Кредит (68, 30)  
 Конец Если  
 Если Метод определения выручки По Оплате Тогда  
 Начисляем неоплаченный при отгрузке НДС.  
 Проводка Дебет (76, 4) Кредит (68, 2)  
 Если в документе указан НГСМ Тогда  
 Начисляем неоплаченный при отгрузке НГСМ.  
 Проводка Дебет (76, 4) Кредит (67, 2)  
 Конец Если  
 Конец Если  
 Проводка Дебет (50) Кредит (62)  
 Конец обработки счета 62

## Сравнение текста и дракон-схемы

Ясно, что схема на рис 142 намного наглядней, чем текст. Почему?

Правила языка Дракон заставляют разбивать графическую информацию на отдельные обособленные блоки. Двумерное пространство графики позволяет наглядно расположить эти блоки и показать связь между ними.

В тексте мы используем практически одно измерение — вертикаль. А в дракон-схеме информация эффективно разворачивается в двух измерениях — вертикальном и горизонтальном. Это очень важно, потому что система «глаз—мозг» лучше воспринимает панорамную информацию.

## Дракон решает проблемы

Когда мы только начинали использовать в своей работе дракон-схемы, то опасались непонимания бухгалтеров, настраивались на долгие объяснения. К нашему удивлению, вопросов о внутреннем устройстве дракон-схем практически не было. Многие бухгалтеры, глядя на Дракон-схемы, сразу же выделяли существенные для себя проблемы. И тут же начинали объяснять, что их не устраивает. И какие изменения следует внести, чтобы учесть особенности конкретного производства. Очень быстро устанавливалась атмосфера взаимного понимания решаемой задачи.

Любой специалист, занимающийся автоматизацией, знает, как трудно понять: что хочет заказчик? Применение языка Дракон дало нам возможность преодолеть непонимание и недоверие бухгалтеров при внедрении программ 1С.

Кроме того, мы используем язык Дракон для проектирования задач бухгалтерского учета. То есть, применяем дракон-схемы, записывая новую информацию при постановке задач «с нуля».

Подведем итоги. После нескольких лет проблем и трудностей, которые мы испытывали при обучении и внедрении бухгалтерских программ, наконец, найдена нужная методика работы. Теперь в наших руках есть набор правил и процедур языка Дракон, с помощью которых можно ясно и полно описать нашу работу.

Самое главное, мы добились, что бухгалтер стал понимать суть работы программы. Язык Дракон решил для нас с заказчиком важнейшую проблему взаимопонимания.

*Наши координаты:* ООО «Инженеры информации», город Волжский Волгоградской области тел. 8 (8443) 25-57-83, 8 (8443) 25-97-21

Сайт: <http://www.enginf.ru/>

См. далее две дракон-схемы

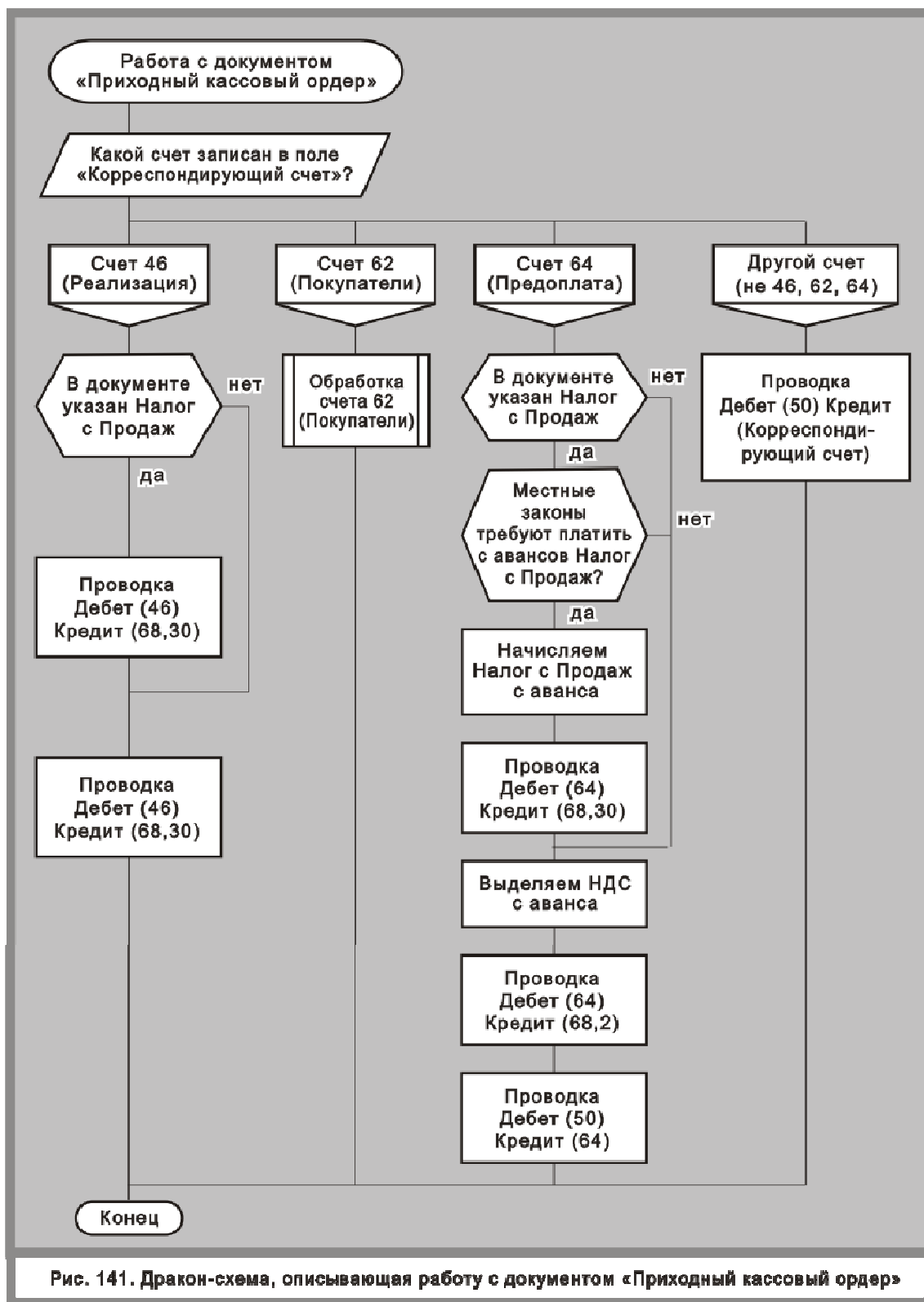


Рис. 141. Дракон-схема, описывающая работу с документом «Приходный кассовый ордер»

