

**В.Г. МАТВЕЙКИН,  
Б.С. ДМИТРИЕВСКИЙ,  
Н.Р. ЛЯПИН**

**ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ  
ВЫПОЛНЕНИЯ БИЗНЕС-ПРОЦЕССОВ  
В СИСТЕМЕ ЭЛЕКТРОННОГО  
ДОКУМЕНТООБОРОТА**

• ИЗДАТЕЛЬСТВО ТГТУ •

УДК 004.032.34(075)  
ББК У290с51я73  
М336

Р е ц е н з е н т ы:

Доктор физико-математических наук, профессор ТГТУ  
*С.Н. Дзюба*

Доктор технических наук, доктор экономических наук, профессор ТГТУ  
*Б.И. Герасимов*

**Матвейкин, В.Г.**

М336 Интеллектуальный анализ выполнения бизнес-процессов в системе электронного документооборота : учебное пособие / В.Г. Матвейкин, Б.С. Дмитриевский, Н.Р. Ляпин. – Тамбов : Изд-во Тамб. гос. техн. ун-та, 2008. – 88 с. – 100 экз. – ISBN 978-5-8265-0713-1.

Рассмотрены на примере реальной системы DocsVision основные аспекты внедрения системы электронного документооборота: сложность и недостатки традиционного подхода; аргументация необходимости внедрения; типовая архитектура; наиболее популярные системы, предлагаемые на отечественном рынке. Представлен алгоритм восстановления комплексных моделей автоматизированных бизнес-процессов, алгоритмы поиска частых подпоследовательностей, описаны сети Петри.

Предназначено для студентов специальности 080801 "Прикладная информатика в экономике", может быть использовано при курсовом и дипломном проектировании.

УДК 004.032.34(075)  
ББК У290с51я73

**ISBN 978-5-8265-0713-1**

© Матвейкин В.Г., Дмитриевский Б.С.,  
Ляпин Н.Р., 2008  
© ГОУ ВПО "Тамбовский государственный  
технический университет" (ТГТУ), 2008

В.Г. Матвейкин, Б.С. Дмитриевский, Н.Р. Ляпин

**ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ  
ВЫПОЛНЕНИЯ БИЗНЕС-ПРОЦЕССОВ  
В СИСТЕМЕ ЭЛЕКТРОННОГО  
ДОКУМЕНТООБОРОТА**

*Утверждено Ученым советом университета  
в качестве учебного пособия для студентов специальности  
080801 "Прикладная информатика в экономике"*



---

Тамбов  
• Издательство ТГТУ •  
2008

Учебное издание

МАТВЕЙКИН Валерий Григорьевич,  
ДМИТРИЕВСКИЙ Борис Сергеевич,  
ЛЯПИН Никита Романович

**ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ВЫПОЛНЕНИЯ  
БИЗНЕС-ПРОЦЕССОВ В СИСТЕМЕ ЭЛЕКТРОННОГО  
ДОКУМЕНТООБОРОТА**

Учебное пособие

Редактор З.Г. Чернова  
Инженер по компьютерному макетированию М.Н. Рыжкова

Подписано к печати 03.06.2008  
Формат 60 × 84/16. 5,11 усл. печ. л. Тираж 100 экз. Заказ № 286

Издательско-полиграфический центр ТГТУ  
392000, Тамбов, Советская, 106, к. 14

## ВВЕДЕНИЕ

От организации работы с документами зависит эффективность работы всего предприятия. Документы несут в себе информацию о внешней среде предприятия, что очень важно с точки зрения принятия решений. Внешняя среда характеризуется сложностью, подвижностью (скоростью изменения различных компонентов и факторов в окружении организации) и неопределенностью. Причем для разных функциональных зон скорости изменения, как и объем факторов внешней среды, не являются одинаковыми. Неопределенность внешней среды является функцией количества информации, которой располагает организация по поводу конкретного факта, а также степенью уверенности в достоверности информации. Качество принимаемых решений напрямую связано со степенью неопределенности внешнего окружения. Однако определить ресурсы, релевантно обеспечивающие деятельность организации, – чрезвычайно сложная задача [1], особенно в современном информационном обществе.

Компьютеризация офиса – оснащение рабочих мест персональными компьютерами и объединение их в сети – привело в последние годы к интенсивным работам по созданию программного обеспечения автоматизации документооборота и появлению в этой области большого числа проблемно-ориентированных продуктов. Идея, лежащая в основе систем автоматизации документооборота, – выделение функций по обработке документов, которые непосредственно не зависят от характера самих документов и связанных с ними приложений.

Такие функции, как регистрация документов или перемещение их по организации, могут быть реализованы независимо от того, являются ли сами документы служебными записками, договорами или платежными документами. Таким образом, системы автоматизации документооборота являются, по сути, интегрирующими технологиями, обеспечивающими унифицированное управление документами в организации и связанными с ними специфическими приложениями.

Различные классы систем ориентированы на различные модели документооборота. В частности, системы типа групповой работы ориентированы на организацию коллективной работы над документами. В системах типа управления документами основное внимание уделяется построению и ведению документальных баз – справочников над множеством документов. Управление движением документов по учрежденческой сети является основной функцией целой группы продуктов, начиная от простейших систем электронной почты и заканчивая изощренными системами автоматизации деловых процедур.

Непосредственно за организацию делопроизводства, т.е. за реализацию принятых правил обработки документов, на предприятиях, как правило, отвечает специальное подразделение – так называемая служба документационного обеспечения управления. На конкретных предприятиях эта служба может называться управлением делами, канцелярией, секретариатом, общим отделом и т.д. В ведении этой службы обычно находится и делопроизводственный персонал других структурных подразделений предприятия.

Функции делопроизводственного персонала заключаются в приеме и регистрации документов, в продвижении документов по предприятию с фиксацией резолюций и отчетов, рассылке документов вовне, а также контроле исполнения документов (резолюций). Для управления движением и исполнением документов делопроизводственный персонал ведет специальные журналы и/или картотеки, в которых фиксируются перемещения документов, а также связанные резолюции и отчеты исполнителей.

Правила регистрации документов на предприятии, состав регистрируемых реквизитов, правила оформления резолюций и передачи документов, правила контроля исполнения, а также порядок заполнения отчетных форм регламентируются соответствующими инструкциями.

Традиционная технология, описанная выше, имеет принципиальные ограничения:

- Информация о документах и ходе их исполнения распределена по системе картотек организации и ее структурных подразделений. Поэтому получение информации о работе с документами на предприятии требует поиска и обработки данных из разнородных и децентрализованных картотек.
- Картотеки документов обычно отделены от исполнителей. Они содержат неполную и неоперативную информацию о состоянии документов.
- Размножение и перемещение большого количества бумажных документов, ведение многочисленных и дублирующих друг друга журналов и картотек приводит к большим объемам непроизводительных затрат.

Перечисленные проблемы еще более усугубляются в корпоративной системе управления, когда необходимо координировать деятельность организационно самостоятельных структур, территориально удаленных друг от друга.

Именно поэтому в последнее время большой интерес направлен к системам электронного документооборота (СЭД), снимающих описанные выше ограничения и предоставляющих организации принципиально новые возможности. Одним из ключевых преимуществ использования СЭД является наличие информационной базы данных, хранящей в себе всю историю взаимодействия сотрудников и подразделений внутри организации.

В результате длительной работы СЭД могут скапливаться огромные массивы информации, анализ которой обычными средствами – трудновыполнимая задача. В связи с чем в последнее время разрабатываются специальные алгоритмы извлечения знаний, основанные на интеллектуальном анализе данных [2 – 4]. Рассмотрению этих алгоритмов и проблем их использования посвящено данное учебное пособие.

## 1. СИСТЕМЫ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА

### 1.1. СУЩНОСТЬ УПРАВЛЕНИЯ ДОКУМЕНТООБОРОТОМ ФИРМЫ

Первый шаг в управлении документацией – это определить, чем именно мы собираемся управлять. Очевидно, что для понимания этого вопроса необходимо дать определения ряду понятий.

Начнем с понятия "документ". Документ (от лат. documentum – доказательство, свидетельство) – понятие информационное, которое в отдельных случаях (при наличии подписи и печати) дополняется юридическим понятием.

Документ – это любая семантическая информация, которая выражена на любом носителе с целью ее постоянного обращения в динамической информационной системе. Обращение означает последовательную смену информационных процессов: передача, прием, обработка, хранение, поиск, размножение. В категорию документов попадают самые разнообразные

источники: книги, письма, фотографии, кинофильмы. Стоит особо отметить, что документ не обязательно должен существовать в виде записи на бумаге. Это могут быть данные в цифровом представлении, приведенные в наиболее удобный для пользователя формат, например, заказ на поставку, запрос на обновление персонала или чек.

Содержание документа и контекст, в котором документ подается, носят название "предмет документа".

В системах управления документооборотом обязательным является наличие у документа регистрационной карточки – набора реквизитов документа (вид документа, регистрационный номер, краткое содержание и другие атрибуты, в общем случае регламентируемые ГОСТ, но они могут отличаться в конкретных случаях).

Следующий термин, которому необходимо дать определение, – управление, особенно, применительно к документам. Управление – это разработка и поддержка условий, при которых ресурсы, отсортированные по группам, смогут давать должную отдачу и эффективно действовать (использоваться) для достижения общих целей [5].

Управление обычно ассоциируется с деятельностью на уровне группы в связи с устранением проблемы или решением задачи и включает в себя планирование, руководство, организацию, поиск ресурсов и контроль. Кроме того, на любой фирме ведется деятельность по общему управлению, управлению оборудованием, финансами, персоналом, информационными ресурсами, маркетинговыми мероприятиями, операционному и стратегическому управлению.

Исходя из предложенных определений документа и управления, можно сформулировать определение управления документацией – процесс наблюдения и контроля за официальными деловыми бумагами фирмы, записями принятых решений и важными для фирмы промежуточными разработками, представленными в формате документа. При этом не важно существует документ в электронном виде или на бумаге.

## 1.2. БИЗНЕС-ПРОЦЕССЫ УПРАВЛЕНИЯ ДОКУМЕНТООБОРОТОМ

Основные процессы, которые затрагивает система управления документооборотом:

- работа с входящими документами;
- работа с исходящими документами;
- работа с внутренними документами;
- создание маршрута документа;
- создание шаблона документа;
- классификация документов;
- учет сторонних организаций;
- контроль работы с документами.

Каждый из этих процессов состоит из множества подпроцессов, например процесс работы с входящими документами состоит из процесса регистрации, процесса поиска, процесса управления доступом и т.д.

При традиционном бумажном документообороте исполнение документа, включая вынесение по нему резолюции, связано с передачей оригинала документа или его копии одним должностным лицом другому. При этом любое перемещение документа, действие над документом фиксируется соответствующими службами (например, канцелярией) и сотрудниками (делопроизводителями) путем внесения соответствующих записей в регистрационную карточку документа либо регистрационный журнал. С момента регистрации регистрационная карточка документа попадает в картотеку и остается там до списания документа в архив.

Когда возникает необходимость в получении справки о состоянии дел или в возобновлении информации по конкретному вопросу, следует обращение к картотеке, находятся соответствующие карточки документов и на основе внесенной в них информации делаются выводы. Это подчеркивает важность и необходимость быстрого и своевременного доступа к подобному рода картотекам, учитывая, что такой доступ может осуществляться часто и разными сотрудниками одновременно.

Упрощенная схема исполнения поступившего извне документа (входящего документа) приведена на рис. 1.

Из приведенной схемы видно, что при исполнении документа возникает два информационных потока. Первый поток связан с движением самого документа:

- документ поступает в канцелярию, где его регистрируют;
- из канцелярии документ передается руководителю для вынесения по нему резолюции;
- после вынесения резолюции и назначения исполнителей документ и его копии рассылаются исполнителям;
- исполненный документ списывается в дело.

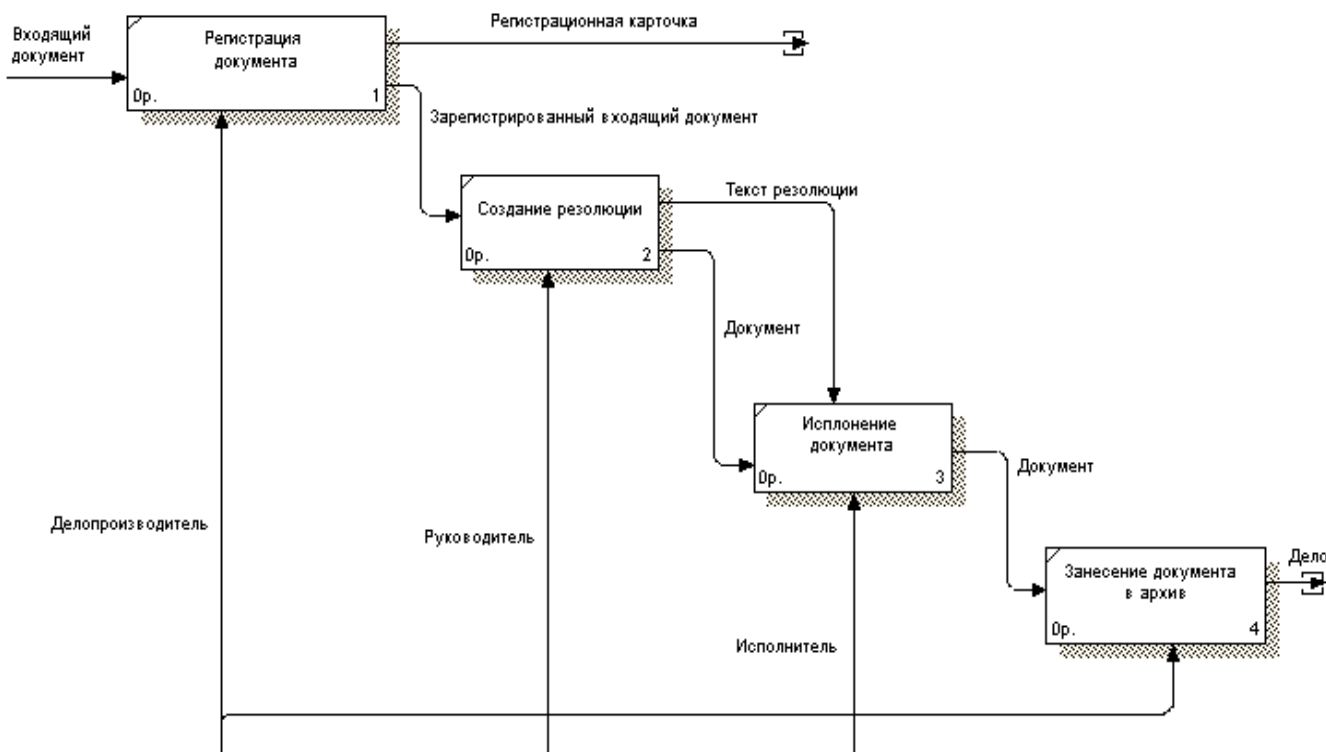


Рис. 1. Схема исполнения входящего документа

Второй информационный поток замкнут на регистрационную карточку документа:

- в момент регистрации в карточку вносятся реквизиты: регистрационный номер, присвоенный документу; информация о корреспондентах поступившего документа; краткое содержание документа; другие данные общего характера.
- факт вынесения резолюции фиксируется путем внесения в карточку информации об авторе резолюции, текста резолюции, фамилии исполнителей, срока исполнения документа, назначенного автором резолюции.
- по исполнении резолюции в карточку добавляются тексты отчетов исполнителей и фактическая дата исполнения документа.

Основным объектом автоматизации СЭД является второй информационный поток, связанный с регистрацией документов, хранении информации о документах и получении различного рода отчетов по данным показателям.

Это особенно актуально, учитывая, что практически на всех крупных предприятиях существуют общие отделы. Их название варьируется – это может быть отдел общего делопроизводства, общий отдел, управление делами и т.д., но самое главное – суть данного подразделения остается одинаковой и заключается она в организации потоков документов. Кроме центрального документооборота крупного предприятия существуют специализированный документооборот (в подразделениях предприятия, в отделах специализированного делопроизводства и т.д.).

Основным принципом документооборота является отсутствие локальных архивов документов. Любой входящий документ, не попавший в общий архив документов, является потенциальным кандидатом на неисполнение. В организации есть соответствующая служба, а именно канцелярия, которая отвечает за прохождение документов. Документы, не учтенные в канцелярии, теряются чаще всего.

В организации должен существовать только один канал поступления входящих документов. Даже если прием документов построен по распределенной схеме, то это сделано только для повышения скорости обработки документов. Все такие подразделения мгновенно предоставляют информацию в канцелярию (желательно в режиме on-line связи с канцелярией). Можно рассматривать, что такие удаленные пункты регистрации документов являются подразделениями канцелярии.

Для полноты картины документооборота на предприятии система документооборота не должна заканчиваться на ответственных исполнителях, т.е. система автоматизации документооборота организации не должна замыкаться только на канцелярии, а должна иметь продолжение в системах управления документооборотом подразделений организации.

Типов документов, используемых в работе предприятия, достаточно много (на отдельных предприятиях их число достигает до 500-600). В то же время, с точки зрения канцелярии, основных видов документов бывает весьма небольшое количество, а именно всего-навсего три вида:

- *Входящие.* Это документы, которые поступили на предприятие от внешних партнеров. Большинство входящих документов должны порождать соответствующие исходящие, причем в заранее установленные сроки. Сроки устанавливаются или нормативными актами, предписывающими то или иное время ответа на соответствующий входящий документ, или сроком исполнения, указанным непосредственно во входящем документе.
- *Исходящие.* Большинство исходящих документов являются ответом организации на соответствующие входящие документы. Некоторая часть исходящих документов готовится на основе внутренних документов предприятия. Небольшое число исходящих документов может требовать поступления входящих документов.
- *Внутренние.* Данные документы используются для организации работы предприятия. Через канцелярию проходят не все внутренние документы, а только переписка наиболее крупных структурных подразделений предприятия (особенно если они территориально разнесены) и приказы руководства предприятия. Так же через канцелярию проходят внутренние доку-

менты, порождающие исходящие. В частности, по общим правилам делопроизводства единственный способ отправить запрос, письмо или материалы во внешнюю организацию это направить внутренний документ в канцелярию, где его преобразуют в исходящий и отправят в стороннюю организацию.

Документы каждого из этих видов могут быть достаточно разнообразны. Это могут быть письма, распоряжения, циркулярные указания и т.д. Обычно под типом документа на предприятии понимается именно эти деления, причем еще более детализованные. С точки зрения канцелярии данное деление достаточно неинтересно, хотя во внутренней полной системе делопроизводства оно безусловно необходимо.

Прежде всего, все документы, проходящие через общий отдел, обладают уникальным регистрационным номером (возможны свои алгоритмы построения номеров для каждого из типов документов). Мало того, любая бумага, не имеющая регистрационного номера, – это просто бумага, а не документ. Документом ее делает именно наличие на ней регистрационного номера.

Атрибуты документов у каждого партнера весьма разнообразны. В то же время можно выделить общую часть атрибутов, которые встречаются практически во всех организациях. Причем эти общие атрибуты несколько различны у всех типов канцелярских документов.

Основными атрибутами канцелярского документа являются:

1. Регистрационный номер документа. Данный номер однозначно позволяет сослаться на документ, прошедший через канцелярию. Структура регистрационного номера может быть различна для исходящих, входящих и внутренних документов. Структура регистрационного номера определяется в каждой организации самостоятельно.

2. Источник документа (контрагент). Указывает откуда получен документ. Для входящих документов это сторонняя организация, для внутренних и исходящих документов это или подразделение или конкретное лицо из руководства предприятия.

3. Ответственный исполнитель документа. Указывает сотрудника предприятия, которому поручено исполнение данного документа или который разработал данный документ (для исходящих и внутренних). Исполнитель документа всегда один и только один. Иногда встречаются два исполнителя документа (для документов длительного исполнения при смене кадрового состава предприятия).

4. Код документа по номенклатуре дел предприятия. Данный атрибут относит документ к тому или иному кругу типовых вопросов, решаемых предприятием в своей производственной деятельности. Номенклатура дел – это формальный список дел предприятия, который утверждается заранее на определенный период времени (обычно на год). Ранее данная номенклатура являлась достаточно статичной, в современных условиях номенклатура дел является высоко динамичным документом, который может обновляться 1 – 4 раза в год.

Далее в документах могут быть атрибуты, специфические для данного типа документа.

Внутренние документы помимо стандартных могут обладать следующими атрибутами:

- Список подразделений предприятия для ознакомления.
- Контрольный срок ознакомления или исполнения.
- Список исполнителей документа.

Исходящие документы имеют дополнительные атрибуты:

• Документ-основание. Этот атрибут есть всегда, так как исходящий документ всегда порождается на основе или входящего или внутреннего.

- Список рассылки.

• Контрольный срок ответа. Данный атрибут встречается редко и не во всех организациях.

В конкретных организациях данные атрибуты могут носить различные наименования или добавляться дополнительными атрибутами, часто большим количеством. Но перечисленные выше атрибуты присутствуют во всех канцеляриях предприятий.

Все документы, проходящие через канцелярию, являются связанными документами в том смысле, что большинство из них ссылается на другие документы. Наиболее типичным случаем является входящий документ, который практически всегда порождает соответствующий ему исходящий.

Без связей как таковых могут появляться только внутренние и входящие документы. Причем входящие документы могут иметь связи как на исходящие, которые вызывают их появление, так и на другие входящие. Все документы связаны как в системе управления документами, так и в системе контроля исполнения (как принадлежащие одной работе). В этом смысле здесь наблюдается некоторое дублирование связей.

Связи в большинстве случаев направлены по принципу "главный-подчиненный". Иногда встречаются ненаправленные связи, которые объединяют родственные документы (документы, посвященные одному вопросу).

В принципе в крупной организации с большим документопотоком работа канцелярии может быть построена по децентрализованному принципу. При этом каждое подразделение предприятия получает свою долю корреспонденции, относящуюся к ней. Номера входящей корреспонденции создаются либо в едином нумераторе, либо в номер входящего документа включается номер подразделения, получившего документ, а нумераторы в каждом подразделении ведутся независимо. Исходящая корреспонденция обычно всегда проходит через одно место (по крайней мере, пока не известны предприятия с другой схемой обработки исходящей корреспонденции).

Организация децентрализованного приема входящих документов достаточно распространена и во всем мире. Достаточно привести пример газеты Times. При реорганизации отдела обработки писем граждан в конце 1980-х гг. (тоже канцелярия, но специализированная) значительная часть работы по сортировке корреспонденции была передана Британскому почтовому ведомству. Для этого газета стала принимать корреспонденцию в несколько абонентских почтовых ящиков (до этого она имела один адрес для всей корреспонденции). После двух лет эксплуатации выяснилось, что число ошибок в адресах (отправлено не в тот абонентский ящик) составляет меньше 2%. Данная реорганизация позволила сократить количество сотрудников отдела обработки входящей корреспонденции в два раза. Это яркий пример, что процесс анализа документопотоков может принести значительный результат не только если он заканчивается полной автоматизацией предприятия.



В работе канцелярии наиболее "узким" местом остается работа с письмами граждан. В большинстве случаев подобная работа относится к общему делопроизводству. Исключение составляют организации, для которых работа с письмами составляет основную производственную деятельность (телерадиокомпании, газеты, журналы, информационные агентства и другие средства массовой информации). При обработке писем граждан применяются несколько другие классификаторы. Важна развернутая статистика по атрибутам писем, что основывается на опыте разработки автоматизированной системы обработки корреспонденции Гостелерадио России [2]. В крупных государственных организациях тоже существует специальный учет писем граждан и строгий контроль за их выполнением. Таким образом, при внедрении СЭД блок обработки писем граждан нуждается в дополнительном исследовании и описании.

Основным результатом работы канцелярии являются отчеты, которые она предоставляет руководству предприятия о потоке. Отчеты условно можно разделить на две большие группы:

- *Оперативные.* Выводятся ежедневно или еженедельно. Основное их назначение – получить объективную картину информации для оперативного управления.
- *Аналитические.* Они служат для анализа общей картины документооборота на предприятии. Типичные периоды вывода отчетов данной группы – месяц, квартал, полугодие, год.

Рассмотрим более подробно оперативные отчеты. Основное их содержание – список работ, которые должны быть завершены в ближайшее время. Получатели отчетов – ответственные исполнители и руководство предприятия. Отчеты бывают сводными и индивидуальными. Индивидуальные отчеты представляют собой сводку работ, находящихся в компетенции того или иного должностного лица. Сводный отчет – это список работ, находящихся в стадии завершения на заданный период времени (возможно разделенный на отдельные подразделы по различным признакам). Сводные отчеты поступают к руководству предприятия или к руководству канцелярии для общего контроля за процессом делопроизводства. Индивидуальные отчеты поступают непосредственно к исполнителям или контролирующим лицам.

При рассмотрении отчетов в СЭД их группировка не производится. У каждого оперативного отчета имеется фильтр, задающий интервал времени, в который происходит окончание работ. При этом данные фильтра помещаются в заголовок отчета. Типичными примерами отчетов являются:

1. Завершающиеся работы по исполнителям.
2. Сводный отчет по подразделениям организации.
3. Сводный отчет по контрагентам.

Рассмотрим подробнее аналитические отчеты. Хотя существует достаточное количество стандартных форм отчетов, каждая организация может заводить собственные формы отчетов (в том числе и матричные). Данные отчеты служат для планирования работы организации в дальнейшем, анализа номенклатуры предприятия, списка контрагентов и т.д. Часто аналитические отчеты выдаются по отдельным структурным подразделениям организации. Отчеты представляют картину работы предприятия за определенный период времени. Следовательно, в фильтр представления отчета в СЭД вводится параметр – время анализа. В нем можно задавать либо произвольный, либо фиксированный интервал времени:

- а) за месяц (указывается месяц и год, по умолчанию текущий);
- б) за квартал (по умолчанию текущий квартал);
- в) за год (по умолчанию текущий год).

Аналитические отчеты выдаются сравнительно редко. Часто их оформляют в виде отдельного издания и распространяют по всей организации. В связи с этим к ним предъявляются достаточно высокие требования по качеству оформления. В общем случае они должны включать логотипы предприятия, гербы и прочую графическую символику. Такие отчеты для исполненных документов могут быть представлены в виде списка документов по исполнителям, по контрагентам или по номенклатуре.

Канцелярия крупного предприятия является центром документооборота, но на ней документооборот не заканчивается. Она тесно связана с документооборотом подразделений предприятия, с системами поддержки принятия решений и с архивом предприятия. В общем случае можно рассматривать систему автоматизации документооборота как центральную систему, осуществляющую координацию остальных подсистем документооборота, своеобразного ядра системы автоматизации производственной деятельности предприятия.

На одном уровне с канцелярией стоит архив предприятия, который служит для накопления документов долговременного хранения и их поиска.

Все документы, проходящие через канцелярию, попадают в архив. Рассмотрим организацию документов в архиве предприятия.

Все документы подразделяются на дела. Одно дело объединяет все документы за один год работы предприятия по тому или иному пункту номенклатуры дел организации. Объем единицы хранения в архиве не может превосходить 200 страниц. В связи с этим для больших дел вводятся несколько томов. Таким образом подразделение документов в архиве происходит по делам, а затем по томам. Поиск информации осуществляется как по атрибутам дел и томов, так и по атрибутам документов.

Дела в архив передаются из канцелярии либо отдельно (по томам), либо целиком по окончании года. В любом случае дело закрывается в конце года. При этом проверяется полный перечень документов в деле, их разбивка по томам. После этого готовится описание дел, в которой указывается список дел (по их наименованию), количество томов в деле, количество страниц в каждом томе. После составления описи дел за конкретный год они поступают на хранение.

Сроки хранения для разных позиций номенклатуры различны. Типовые сроки хранения документов в крупной организации: 1, 3 года и 5, 10, 30, 50 лет. После истечения срока хранения дела оно из архива предприятия передается на хранение в государственный архив (для государственных организаций). Для частных компаний после истечения срока хранения дела уничтожаются. И тот и другой процесс оформляется соответствующими актами. Дела в архив или на уничтожение поступают как единое целое, т.е. в отличие от процесса формирования архивных дел запускается соответствующая операция для всех томов дела.

Документооборот на предприятии не заканчивается только канцелярией. Мало того, он только начинается в канцелярии, а основная часть документооборота проходит через подразделения. В принципе канцелярию за редким исключением не волнует документооборот в отделах.

На уровне подразделений СЭД должна позволять вести внутренний документооборот и передавать дела в канцелярию и, возможно, в другие подразделения предприятия. Документооборот уровня отдела отличается от документооборота уровня подразделения тем, что дела свободно ходят в рамках подразделения. Переходы дел между отделами внутри одного подразделения не регистрируются, но выход дела за рамки отдела должен отслеживаться и регистрироваться.

Исходя из всего описанного выше, мы можем дать следующее определение. Цель систем управления документооборотом – распределить общие информационные ресурсы компании таким образом, чтобы они оставались под надежной защитой, и чтобы их можно было найти, получить или переслать, а также предоставить возможности по маршрутизации документов. Распределение документов должно быть возможным независимо от формы их представления – на бумаге или в электронном виде.

### 1.3. НЕДОСТАТКИ ТРАДИЦИОННОГО ПОДХОДА В УПРАВЛЕНИИ ДОКУМЕНТООБОРОТОМ

Хранение и поиск документов являются очень важным видом деятельности для фирмы, от чего зависит в наши дни выживание компании. Стало очевидно, что традиционный бумажный подход к ведению записей и управлению документооборотом устарел и малоэффективен. Можно выделить ряд причин, по которым имеет право быть признанным верным:

- *Несоблюдение сотрудниками фирмы установленных норм при работе с документами.* Человек, получивший документы, важные для его деятельности, склонен как правило, припрятывать информацию для себя. В лучшем случае закончив выполнение задания, он сложит все документы, имеющие отношение к проекту в одну папку.

- *Вопрос безопасности.* Увеличение количества создаваемых документов, публикаций и возможность распространения информации по электронной почте или локальным сетям фирм привели к выходу на первый план проблем защиты документов, контроля, отслеживания и поиска. Раньше папки сортировались по значимости наиболее важного документа в каждой из них, чтобы защитить информацию и ограничить доступ к ней. В наши дни системный администратор, имеющий соответствующий его статусу права доступа, может прочитать любое письмо, посланное по электронной почте в пределах доверенной ему сети.

- *Проблема эффективного поиска.* Действующая в большинстве организаций система поиска документов базируется на присвоении папкам с файлами заголовков или на списках ключевых слов, описывающих общее содержание файла, что отнюдь не гарантирует точности при поиске документа в файле. В операционной системе MS-DOS установлено ограничение на длину имени файла или директории – она не должна превышать восьми символов, что еще больше обострило проблему. В MS-Windows это ограничение снято, но проблема, тем не менее, не решена, так как не существует единых стандартов наименования различных документов. Системы электронной почты обычно сохраняют послания в пределах структуры своей собственной базы данных, тем самым ограничивая возможности поиска по письмам с помощью поискового механизма для текстовых документов.

- *Фактор недостаточного использования имеющихся в организации компьютеров.* Компьютеры в большей своей массе продолжают использоваться как усовершенствованные печатные машинки, и авторитетные специалисты не раз отмечали, что внедрение вычислительных средств без серьезной проработки информационной структуры предприятия лишь усиливает неразбериху. Использование компьютеров без систем управления документооборотом способствует в гораздо большей мере росту "электронного" документационного хаоса, чем это случается с бумажными документами. И, тем не менее, компьютеры, а в частности компьютерные сети, не используются в полную силу своих возможностей, т.е. у предприятий есть потенциал, который простаивает.

- *Более длительное время существования электронного носителя документа.* Имеющиеся в настоящее время магнитные и оптические носители имеют оценочное время жизни около 10 лет.

- *Себестоимость электронного документа меньше.* Вычислительные мощности персональных компьютеров, рабочих станций и серверов (как процессора, так и памяти) удваиваются каждые 18 месяцев. Таким образом, компьютерные ресурсы по памяти и по производительности увеличиваются быстрее, чем технические достижения, которые удваиваются каждые 10 лет (7 % в год). Оборудование для хранения информации сейчас на много дешевле, чем это было 10 – 20 лет назад: можно купить многогигабайтный магнитный носитель за несколько тысяч долларов. CD-ROM'ы стоят порядка нескольких долларов.

У электронных документов выше интерактивность. Бумажная среда для документов является средой негибкой, консервативной. После того, как документы напечатаны, остается очень мало возможностей для их изменения. Исправления к бумажным документам обычно имеют вид указания на ошибку, которое публикуется в последующих изданиях. Наоборот, электронные документы могут непрерывно модернизироваться автором. В печатной продукции нет удобной возможности поддерживать читательские комментарии, примечания и авторские исправления в ближайшее реальное время, что вызвано более длинным циклом исправления и распространения.

### 1.4. ТЕХНОЛОГИЯ ПОТОКОВ РАБОТ КАК ПРИНЦИПИАЛЬНО НОВЫЙ ПОДХОД К УПРАВЛЕНИЮ ПРЕДПРИЯТИЕМ

Workflow – автоматизация полностью или частично бизнес-процесса, при которой документы, информация или задания передаются для выполнения необходимых действий от одного участника к другому в соответствии с набором процедурных правил.

Система управления workflow – система, которая описывает поток работ (по сути, бизнес-процесс), создает его и управляет им при помощи программного обеспечения, которое способно интерпретировать описание процесса, взаимодействовать с его участниками и при необходимости вызывать соответствующие программные приложения и инструментальные средства. Таким образом, система workflow автоматизирует процесс, а не функцию. Появление ее и соответствующих программ-

ных средств workflow – это реакция рынка информационных технологий на внедрение новых принципов в управление предприятиями.

Система workflow автоматизирует процесс, а не функцию. Практически все предыдущие решения в области управления (чаще всего реализованные в технологиях СУБД) позволяли достаточно эффективно автоматизировать отдельные операции и функции, а не процесс (например, функцию продаж, которая является частью процесса обслуживания (клиента)). В рамках этих решений сотрудники, сидя за своими компьютерами (или терминалами), обмениваются информацией с базами данных и между собой, получают цифры, справки, документы, формируют отчеты. При этом последовательность действий сотрудников и правила их взаимодействия определены в лучшем случае инструкциями, а за правильностью их выполнения следит вышестоящее начальство, но информационной системой это никак не поддерживается.

Процессный подход заставил менеджмент предприятий сконцентрировать внимание именно на правилах и взаимодействиях участников процесса, так как эти аспекты являются основными центрами потерь в силу своей размытости и неопределенности. Необходимость в рамках автоматизации отдельных функций иметь средства для автоматического отслеживания последовательности и времени их выполнения, маршрутов документов, занятости сотрудников на различных стадиях процесса привели к идее создания систем класса workflow.

При внедрении workflow-системы предприятие получает следующие преимущества:

- Workflow усиливает контроль над производительностью выполнения задач, связанных с информацией. Повышая конфиденциальность и ужесточая контроль доступа, workflow одновременно привносит "промышленные" методы руководства и управления процессами.

Улучшение качества обслуживания, повышение его оперативности, четкая информация о состоянии запроса, и упрощение доступа к представителям компании. Каждый работающий видит перечень функций, которые он должен выполнить, и может организовать свою работу соответствующим образом. Можно наглядно представить контекст каждой функции. Для сотрудников workflow означает гибкость в работе, быстроту исполнения и высокую степень комфорта. Workflow позволяет принимать решения в нужный момент и представляет достаточную информацию, чтобы руководство могло эффективно вмешиваться в процесс. Workflow дает возможность менеджерам действовать более оперативно, быстрее и компетентнее, обеспечивая постоянный доступ к информации о состоянии каждого заказа, а система мониторинга позволяет держать ситуацию под контролем. С помощью workflow можно сделать более эффективной функцию контроля, существенно приблизив ее к промышленному аналогу.

Автоматизация процедур на базе workflow предоставляет в распоряжение организационных аналитиков всю необходимую статистику для анализа рабочих нагрузок, затрат, периодов пиковой нагрузки и многих других аспектов деятельности компании.

Использование инструментальных средств в рамках workflow позволяет моделировать процедуры и возможные сценарии их выполнения с беспрецедентной степенью детализации и точности, а доступ к данным о выполнении процессов требует минимальных затрат.

## 1.5. СПИСОК СТАНДАРТОВ В ОБЛАСТИ СИСТЕМ УПРАВЛЕНИЯ ДОКУМЕНТООБОРОТОМ

*Стандарты и спецификации, применяющиеся в системах управления документацией:*

- Сервисы сети:

IEEE P1003.x: Управление файлами (NFS);

ISO 9594 (ITU X.500): Сервис директорий;

ITU X.400: Система обработки сообщений (MHS).

- Сервисы систем управления информацией:

UN/EDIFACT (ANSI X12): Обмен данными в электронном представлении (EDI);

ISO 9579: Удаленный доступ к базам данных (RDA);

ISO 9075: Структурированный язык запросов (SQL);

ISO 10027 (ANSI X3.138): Словарная система информационных ресурсов (IRDS);

ISO 8824/8825 (ANSI X.208): Общая система синтаксиса (ASN).

- Сервисы приложений:

ISO 8571: Пересылка, доступ и управление файлами (FTAM);

ISO 9040/9041: Услуги виртуального терминала (VTS).

- Сервисы документов:

ISO 10166: Хранение и поиск документов (DFR).

*Второстепенные стандарты систем управления документооборотом:*

Стандарты архитектуры и представления документов:

ISO 8613. Система стандартов: архитектура учреждений документов и формат обмена (ODF);

ISO 8879: Стандартный обобщенный язык разметки (SGML) и его составная часть язык гипертекстовой разметки (HTML);

ISO 9069: Стандартный обобщенный язык разметки (SGML). Формат обмена документами (SDIF);

ISO 10744: Гиперсреда и синхронизированные структурированные документы (HyTime);

ISO 10179: Язык обработки данных – язык определения стилей отображения и описания документов (DSSSL);

ISO 10180: Язык стандартного описания страницы (SPDL).

*Графические стандарты:*

ISO 7942: Система графического ядра (GKS);

ISO 8651: Метафайл компьютерной графики (CGF);

Объединение экспертов по фотографии (JPEG);

Группа экспертов по движущимся изображениям (MPEG).

*Кодировка шрифтов:*

ISO 10646: Unicode – Универсальное многобайтное кодирование символов;

ISO 9541: Обмен информацией о шрифтах;

ISO 646: ISO Шрифт для обмена информацией в 7-битной кодировке;

ISO 2022: ISO Шрифт в 7-битной и 8-битной кодировке – техника расширения кодов;

ISO 8888: Стандарт дата/время;

ISO 8601: Формат дата/время.

*Спецификация сервисов библиотек фирм-производителей:*

DMA: Единый стандарт управления документацией, представленный Ассоциацией управления информацией и изображениями, а также объединенный стандарт Сети работы с документами (DEN – Document Enable Networking) и моделей взаимодействия хранилищ по Шемроку (Shamrock standards for repository interoperability models);

ODMA: Интерфейс прикладного программного обеспечения для открытого управления документацией;

PDF: переносимый формат документации (собственность компании Adobe);

OpenDoc: стандарт составного документа (собственность компании Component Integration Laboratories).

*Стандарты поиска:*

ANSI Z39.19: руководство по созданию, поддержке и использованию тезауруса;

ANSI Z39.19: стандартный язык запросов;

ANSI Z39.19: обобщенный язык команд (CCL);

SFQL: стандарт структурированного полнотекстового запроса компании Air Transport Association (ATA)/Aerospace Industries Association(AIA).

## 1.6. ОПИСАНИЕ АРХИТЕКТУРЫ СИСТЕМЫ УПРАВЛЕНИЯ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА

При разработке и использовании системы управления документами оговаривают три составные части архитектуры системы: концепцию, логику и предметное воплощение. Эти аспекты соответствуют, хотя и с определенными отличиями, аспектам методики разработки баз данных, согласно которым разработчик создает иерархию, иллюстрирует модель БД (рис. 2).



Рис. 2. Составные части архитектуры системы управления документами

На уровне концепции решаются вопросы масштабности системы управления документами и ее интеграцию в общую систему работы фирмы. Устанавливается взаимосвязь между причинами внедрения системы управления документами и ее пользователями. Первая функция концепции – доказать необходимость внедрения системы управления документами на предприятии, вторая – передать общую идею системы управления документами конечным пользователям.

На логическом уровне решаются вопросы функциональных характеристик системы управления документами (ввод и вывод данных, их обработка, безопасность, правила оформления дел, оформление экрана, форма отчетов). Исходя из пользовательской схемы и интерфейса взаимодействия компонентов, определяется, что будет делать система и когда должен запускаться каждый из процессов. Логика построения системы управления документами не имеет привязки к конкретной системе, на которой она будет запущена, т.е. логическое построение может быть воплощено на любом программном и аппаратном обеспечении. Разработка логической схемы системы управления документами необходима для определения процессов и разделения их на выполняемые машинами и людьми, а также для информирования пользователей о функциях системы управления документами. Базовая схема логического уровня выглядит как показано на рис. 3.

Система управления документами	
Система управления базами данных	
Система управления полнотекстовыми документами	
Менеджер базы безопасности	Менеджер административной базы
Менеджер рабочего процесса	Менеджер гиперссылок

**Рис. 3. Логическая модель архитектуры системы управления документами**

Система управления документами принимает необработанный документ и размещает его в хранилище документации. Функции системы управления документами включают в себя прием и выдачу документов, отслеживание повторов, архивирование, удаление документов и восстановление их из архива или области удаления. Обновлением версии документов также занимается система управления документами.

Система управления базами данных отвечает за все системные данные о документе, которые хранятся в отведенной для этого области памяти. Все важные атрибуты документа (дата создания, автор, тема, ключевые слова, дата внесения в базу и т.д.) проиндексированы. Работа с базами данных ведется посредством мощной, единой для всей системы, СУБД базирующейся на Структурированном языке запросов.

Система управления полнотекстовыми документами индексирует документ по определенным правилам. Эта система часто встраивается в систему, но не является обязательной.

Менеджер административной базы отвечает за функции, позволяющие администратору базы данных или системному администратору создавать резервные копии, восстанавливать, настраивать, отключать, перестраивать и перезапускать различные приложения.

Менеджер базы безопасности отвечает за безопасность системы. Он устанавливает ограничения на доступ к системе определенных пользователей или групп пользователей или ограничивает общий доступ к определенным файлам и директориям.

Менеджер гиперссылок контролирует гиперссылки внутри документов и между ними. Он также отвечает за все объектные связи в составных документах, т.е. в документах, состоящих из нескольких частей, относящихся к разным приложениям.

Менеджер рабочего процесса координирует процесс взаимодействия отделов фирмы между собой и с клиентами компании.

Под предметным воплощением понимается физическая конструкция, обеспечивающая функционирование системы. Принятие решений относительно того, как и где будет внедряться система, происходит применительно к конкретной фирме. Принимается во внимание конфигурация программного и аппаратного обеспечения. Система управления документами в предметном воплощении должна работать в рамках ограничений конкретной системы, т.е. с учетом таких условий, как максимальное число пользователей, объем оперативной памяти, доступность жесткого диска. Более того, на уровне физического воплощения решаются вопросы управления работой системы; например, должен отслеживаться баланс загрузки; проверка данных; создание резервных копий и перезапуск; размещение двоичных программ, файлов, протоколов; файловая структура; сбор статистики.

### **Вопросы для самоконтроля**

1. Дайте определение понятиям документ и управление (применительно к документообороту).
2. Перечислите основные бизнес-процессы, в которых участвует СЭД.
3. Назовите основные типы отчетов, предоставляемых по документообороту.
4. Каковы основные недостатки традиционного подхода к управлению документооборотом?
5. Основные преимущества использования системы управления потоками работ (workflow системы, WFMS).
6. Назовите ключевые аспекты архитектуры системы электронного документооборота.

## **2. СИСТЕМА ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА DOCSVISION**

В этой главе подробно рассматривается реальная отечественная система электронного документооборота DocsVision.

Платформа DocsVision является той основой, на которой организация может построить свою систему документооборота, создавая собственные приложения или используя уже готовые решения:

- Приложение "Делопроизводство", предназначенное для автоматизации делопроизводства в организации, в том числе перехода от обработки бумажных документов к работе с их электронными версиями;
- Приложение "Управление процессами", предназначенное для автоматизации бизнес-процессов, таких как согласование договоров, ознакомление с документом определенного круга лиц и т.д.

Основным инструментом работы пользователя с системой DocsVision является Навигатор, принцип работы которого аналогичен принципу работы Проводника. В Навигаторе отражается дерево папок, в общем случае – сложной структуры, которое по назначению и принципу работы также аналогично дереву папок, которое можно видеть в Проводнике, однако в отличие от Проводника папки Навигатора содержат не файлы, а карточки. Каждая карточка представляет собой максимально подробное описа-

ние отдельного документа – тип; дата создания; фамилия лица, зарегистрировавшего документ; сведения о содержании; ссылки на другие документы, – а также может включать в себя графический файл с изображением отсканированного бумажного документа и файл, содержащий распознанный текст документа. Каждый из указанных выше элементов содержится в отдельном поле карточки.

Карточки размещаются в папках в соответствии с установленными в организации принципами, однако чаще всего дерево папок строится в соответствии со структурой организации, наглядно отображая иерархическую подчиненность ее подразделений, а также принадлежность каждого сотрудника к определенной структурной единице. Так, например, личные папки сотрудников какого-либо отдела обычно находятся в папке с названием отдела.

Кроме этого, во избежание многократного повторного ввода одной и той же информации, а также исключения возможных ошибок в платформе DocsVision предусмотрены справочники (хранящиеся в особой папке *Справочники*) – каталоги различного назначения, содержащие информацию, сгруппированную по определенным критериям. Включаемая в справочники информация далее используется при заполнении некоторых полей карточек документов, позволяя избежать ввода заведомо неверных сведений (например, в тех полях, где должны указываться сотрудники организации – автор документа, исполнитель документа, руководитель подразделения и т.д., – невозможно ввести запись о несуществующем сотруднике), не допуская дублирования информации (например, присваивая вновь создаваемым документам номер в соответствии с принятой в организации классификацией) и избавляя пользователя от рутинной работы по многократному занесению в документы одной и той же информации.

Помимо карточек документов платформа позволяет организовать упорядоченное хранение файлов любого формата, содержащих информацию в текстовой, графической, табличной или другой форме; для этого в механизме платформы предусмотрен особый вид карточки – карточка файла. В зависимости от предполагаемого использования файла в дальнейшем (однократное, постоянное внесение изменений в один и тот же документ – актуализация, использование текста документа в качестве стандартной основы для создания новых документов и пр.) можно выбрать соответствующий способ его хранения.

Для всех типов карточек платформа предусматривает механизм поиска, который может осуществляться по записям карточек, тексту прикрепленных к ним файлов и совокупности условий в любом сочетании.

В зависимости от назначения документации, а также выполняемых сотрудниками обязанностей платформой DocsVision предусмотрена система безопасности, позволяющая разрешать или запрещать сотруднику (группе сотрудников) доступ к любому виду данных в целях ограничения доступа к хранящимся сведениям и гарантии сохранности всей используемой информации. К дополнительным средствам, обеспечивающим безопасность данных, относятся возможности шифрования и/или добавления к документам электронной подписи.

## 2.1. ПРИНЦИПЫ ОРГАНИЗАЦИИ ПЛАТФОРМЫ DOCSVISION

Платформа DocsVision является базисом для создания системы автоматизированного документооборота и включает в себя только основные элементы, предназначенные для хранения информации, доступа к ней, а также обеспечивающие пользователю привычный и удобный интерфейс. Платформа состоит из клиентской и серверной частей.

Серверные компоненты позволяют создать все необходимые структуры для хранения данных и поддерживать их целостность в дальнейшем, обеспечивают получение, обновление и удаление данных клиентом в соответствии с правами доступа, а также многие другие функции.

Клиентские компоненты предоставляют конечному пользователю удобный и понятный интерфейс для осуществления всей работы с информацией, содержащейся в базе данных, включая ее поиск, просмотр и модификацию.

## 2.2. НАВИГАТОР. ОБЩИЕ СВЕДЕНИЯ

Взаимодействие пользователя с системой автоматизации документооборота обеспечивает Навигатор, с помощью которого пользователь системы в соответствии со своими обязанностями в организации и имеющимися правами доступа получает возможность работать со всеми объектами как платформа DocsVision, так и всех остальных приложений, построенных на ее основе.

Основой DocsVision являются карточки, которые группируются по принятым в организации принципам при помощи папок (аналога папок Проводника в Microsoft Windows). Карточка является максимально подробным описанием документа, файла или какого-либо объекта системы DocsVision. Навигатор допускает одновременный просмотр структуры хранения карточек, а также избранную пользователем информацию об их содержимом.

Набор инструментов Навигатора обеспечивает возможность осуществлять все необходимые операции над информацией базы данных, а именно: модифицировать справочники; создавать, удалять и модифицировать карточки документов и заданий; фиксировать выполнение заданий по документам, а также получать доступ к подсистеме маршрутизации, решая, тем самым, основные задачи документооборота – регистрацию и движение документов.

Запуск Навигатора осуществляется аналогично запуску любой другой программы. Главное окно Навигатора состоит из следующих основных элементов:

- Строка меню.
- Панель инструментов.
- Иерархия папок.
- Область просмотра содержимого папки.
- Область предварительного просмотра.

Остановимся подробнее на элементе панели инструментов. Этот элемент содержит кнопки, предназначенные для быстрого доступа к некоторым наиболее часто используемым пунктам меню. Панель инструментов Навигатора может быть отображена/скрыта с помощью установки/снятия флажка в пункте меню **Вид > Панели инструментов > Стандартная**. Назначение кнопок представлено в табл. 1.

## 1. Панель инструментов Навигатора

Название	Назначение
Новая карточка	Используется для создания новой карточки в текущей папке. Тип новой карточки можно выбрать из меню, раскрывающегося при нажатии на правую часть кнопки
Новая папка	Используется для создания в текущей папке дочерней папки
Переместить папку	Открывает окно выбора папки назначения, в которую следует переместить текущую папку
Скопировать папку	Открывает окно выбора папки назначения, в которую следует скопировать текущую папку
Удалить	Используется для удаления объектов
Печать	Позволяет распечатывать данные, отображенные на правой панели Навигатора. При нажатии кнопки открывается окно настройки параметров печати
Обновить	Обновляет иерархию папок и содержимое текущей папки
Свойства папки	Открывает окно свойств текущей папки
Расширенный поиск	Открывает диалоговое окно для задания параметров поиска карточек

Продолжение табл. 1

Название	Назначение
Поиск по представлению	Открывает диалоговое окно для задания параметров поиска текстовой строки в текущем представлении
Поиск папки	Предназначена для поиска папок в иерархии, в название которых входит указанное сочетание символов
Экспорт в Microsoft Excel	Экспортирует в Microsoft Excel данные, отображающиеся в области просмотра содержимого папки
Настройка представлений	Открывает окно настройки представлений
Список папок	Позволяет отобразить/скрыть панель дерева папок
Область предварительного просмотра	Позволяет отобразить/скрыть область просмотра содержимого карточки
Область группировки	Позволяет отобразить/скрыть область группировки
Показать следующую порцию данных	Позволяет вывести следующую порцию данных при ограничении количества выводимых записей в папке

### 2.3. СПИСОК ПАПЕК

Все папки DocsVision организованы в иерархическую структуру (аналогичную дереву папок Проводника Microsoft Windows), которая отображается на левой панели Навигатора. При помощи дерева папок пользователь может получить доступ ко всем карточкам в соответствии с имеющимися у него правами доступа.

В системе предусмотрены следующие типы папок:

1. Пользовательские папки, которые могут создаваться или удаляться:
  - а) Персональная папка пользователя создается администратором при регистрации сотрудника в системе и не может создаваться остальными пользователями.
  - б) Стандартная папка может создаваться любым зарегистрированным пользователем и предназначена для структуризации хранимой информации и большей наглядности при ее просмотре.
  - в) Виртуальная папка, которая предназначена для хранения карточек, отобранных согласно какому-либо определенному критерию.
  - г) Папка-делегат, являющаяся ярлыком на какую-либо другую папку.
2. Системные папки, которые создаются системой автоматически и не могут удаляться пользователями:
  - а) Корневая папка.

- б) Справочники, в которой хранятся все справочники системы.
- в) Результаты поиска, в которой содержатся результаты расширенного поиска карточек.
- г) Корзина, в которую помещаются карточки после удаления.

Пользователь имеет доступ к папкам иерархии в соответствии с предоставленными ему правами; при этом если он не является администратором, ему доступны только Папки, Справочники и Корзина, а также (после занесения записи о нем в справочник сотрудников) его личная папка.

## 2.4. ОБЛАСТЬ ПРОСМОТРА СОДЕРЖИМОГО ПАПКИ

В области просмотра содержимого папки в зависимости от настроек свойств папки по умолчанию может отображаться следующая информация:

- Список находящихся в папке карточек в виде таблицы (*представление*), заголовки которой представляют собой название полей карточек, а строки отражают информацию, содержащуюся в соответствующих полях карточек; набор отображаемых полей карточек зависит от представления, выбираемого в соответствии с потребностями и пожеланиями конкретного пользователя.

- Web-страница, находящаяся по определенному web-адресу.
- Одна из карточек, находящихся в папке.

Пользователь может выбрать способ отображения данных в зависимости от своих текущих задач; настройка способов отображения производится администратором или пользователем с соответствующими правами.

## 2.5. ОБЛАСТЬ ПРЕДВАРИТЕЛЬНОГО ПРОСМОТРА

Область предварительного просмотра позволяет просматривать содержимое карточки, выделенной в списке карточек в Области просмотра содержимого папки, в режиме только чтения без открытия карточки. Функция предварительного просмотра доступна только в том случае, если в свойствах папки выбран способ отображения карточек в виде представления, а также если для данного представления разрешен предварительный просмотр. Область предварительного просмотра может быть отображена/скрыта с помощью установки/снятия флажка в меню **Вид > Область предварительного просмотра** или с помощью кнопки панели инструментов Навигатора *Область предварительного просмотра*.

## 2.6. ОБЛАСТЬ ГРУППИРОВКИ

Группировка позволяет объединять строки представления в группы по каким-либо обоим признакам для отображения информации в более наглядном виде; функция группировки доступна только в том случае, если выбран способ отображения карточек в виде представления. Область группировки расположена на правой панели окна Навигатора над заголовками столбцов таблицы и предназначена для определения и просмотра условий группировки строк таблицы. Максимальное количество одновременной группировки данных по столбцам – 4.

Область группировки может быть отображена/скрыта при помощи установки/снятия флажка в меню **Вид > Область группировки** или с помощью кнопки панели инструментов.

Приведем специфические опции контекстного меню области группировки и ее заголовка:

- Область группировки – позволяет скрыть/отобразить область группировки, содержащую условия группировки.
- Свернуть все группы – оставляет на экране только названия верхних групп и количество входящих в них объектов.
- Свернуть только последнюю группу – позволяет свернуть только самый низкий уровень группировки, очевидно, что если уровень только один, то действие этой опции аналогично действию предыдущей опции Свернуть все группы.
- Развернуть все группы – приводит к отображению на экране всех элементов, распределенных по группам.

## 2.7. РАБОТА С ПАПКАМИ

Персональная папка назначается пользователю администратором при его регистрации в справочнике сотрудников; она предназначена для работы пользователя с документами и хранения ярлыков карточек (например, в приложении "Делопроизводство" при способе маршрутизации *On-line задание* в личной папке пользователя при запуске задания появляется ссылка на карточку задания).

При открытии Навигатора пользователь сразу попадает в свою персональную папку.

Папки, назначаемые сотрудникам в качестве личных (личные папки пользователей), получают тип *Персональная*, указываемый в свойствах папки, где отображается также учетная запись владельца.

Стандартная папка создается для хранения и облегчения поиска карточек; как правило служит для объединения карточек определенного типа, относящихся к отдельной теме или сгруппированных по какому-либо другому признаку. После создания папки пользователь может переопределить ее настройки, сделав их отличными от установленных по умолчанию.

Чтобы создать стандартную папку, необходимо выделить папку в дереве папок. Диалоговое окно создания новой папки можно открыть несколькими способами:

- выбрать в контекстном меню команду **Создать > Папку**;
- нажать кнопку *Новая папка* панели инструментов Навигатора;
- выбрать команду меню **Файл > Создать > Папка**;
- выбрать команду меню **Файл > Папка > Создать подпапку**.

В открывшемся диалоговом окне набирается название новой папки в соответствующем поле и нажмите кнопку *ОК*.

После создания стандартной папки пользователь может изменить ее свойства в соответствии с ее назначением.

Для изменения параметров настройки папки выделяется стандартная папка, свойства которой необходимо изменить, и открывается диалоговое окно *Свойства папки* одним из следующих способов:



- щелкнуть правой кнопкой мыши по нужной папке и выбрать в появившемся контекстном меню команду *Свойства*;
- нажать кнопку *Свойства* панели инструментов Навигатора;
- выбрать в меню последовательно команды **Файл > Папка > Свойства**.

Диалоговое окно *Свойства папки* стандартной папки включает пять вкладок: **Основные свойства**, **Типы**, **Представления**, **Шаблоны** и **Безопасность**; далее уточним свойства папки, настраиваемые на каждой из вкладок.

На вкладке **Типы** (рис. 4) отображается список всех имеющихся в системе типов карточек. Чтобы разрешить создание в стандартной папке карточек только определенных типов, можно установить флажок **Только эти типы карточек могут быть созданы в этой папке** и при помощи флажков отменив в списке нужные типы карточек.

Чтобы разрешить или запретить для определенного пользователя право доступа к какому-либо типу карточек, выделяется интересующий тип карточки с дальнейшим нажатием кнопки *Разрешения*. В открывшемся окне добавляются пользователи и назначаются им права использования указанного типа карточек в данной папке.

На вкладке **Представления** отображается список всех имеющихся представлений. Чтобы разрешить отображение в папке только определенных представлений, устанавливается флажок **Только эти представления могут быть показаны в этой папке** и далее – флажки возле названий представлений, которые могут использоваться для данной папки. Запретить использование любых представлений, кроме дайджеста можно, установив флажок **Только эти представления могут быть показаны в этой папке** и не выбрав ни одного представления.

Разрешить или запретить определенным пользователям право доступа к какому-либо представлению, можно сделать, выделив его и нажав кнопку *Разрешения*. В открывшемся окне добавляется нужный пользователь и определяются его права на использование данного представления в текущей папке.

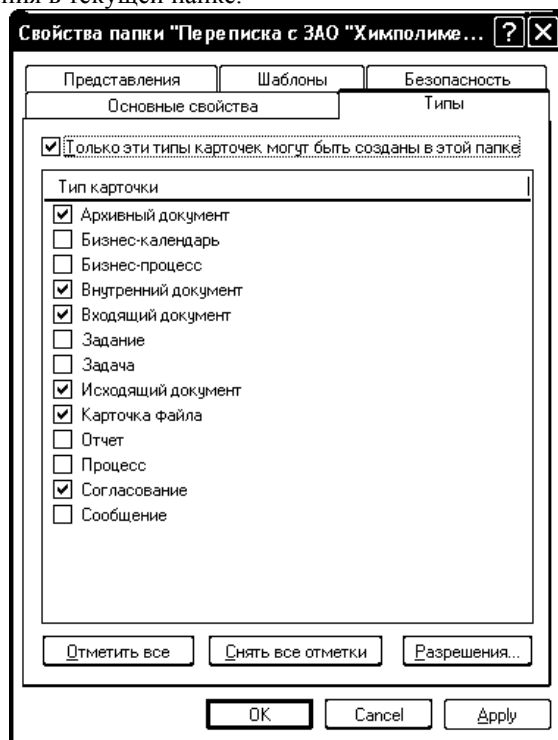


Рис. 4. Диалоговое окно "Свойства папки"

На вкладке **Шаблоны** отображается список всех имеющихся в системе шаблонов для экспорта данных в Microsoft Excel. Чтобы разрешить экспорт данных папки только в определенные шаблоны Excel, устанавливается флажок **Только эти шаблоны могут быть использованы с этой папкой**, а затем устанавливаются флажки напротив названий шаблонов, которые будут доступны для передачи данных папки в Microsoft Excel.

Вкладка **Основные свойства** позволяет изменить настройки отображения карточек, находящихся в папке, сделав их отличными от установленных по умолчанию или указанных при создании папки:

- *Имя папки* – название папки;
- *Тип папки* – Стандартная – информационное поле, не доступное для изменения;
- в поле *Представление* указывается представление, которое будет использоваться для папки по умолчанию; в раскрывающийся список этого поля включаются только те представления, которые были разрешены для использования в данной папке на вкладке **Представления**;
- в раскрывающемся списке поля *Шаблон* выбирается для папки название шаблона Excel, который будет использоваться по умолчанию для экспорта данных папки (набор разрешенных для папки шаблонов определяется на вкладке *Шаблоны*).

В разделе *По умолчанию отображать* выбирается способ отображения содержимого папки по умолчанию:

- *Папку* – табличное отображение карточек, содержащихся в папке;
- *Карточку* – отображение одной определенной карточки (переключатель становится доступным после выбора карточки папки в поле *Карточка папки* с помощью кнопки с многоточием);

- **URL** – отображение Web-страницы, находящейся по указанному адресу; выбирая данный переключатель необходимо указать URL-адрес карточки в поле **URL**.

Следующая группа элементов управления позволяет определить дополнительные настройки свойств папки, служащие для удобства просмотра находящихся в ней карточек.

Флажок **Разрешить использование дайджеста** установлен по умолчанию; его можно снять только в том случае, если на вкладке **Представления** разрешено использование хотя бы одного из существующих представлений. Кнопка *Разрешения* рядом с флажком позволяет назначить определенным пользователям права на использование дайджеста.

Установка флажка **Разрешить автоматическое обновление** приводит к обновлению на экране данных папки через определенный промежуток времени (установленный администратором); в случае снятия флажка обновление данных на экране будет производиться пользователем вручную при помощи кнопки *Обновить* на панели инструментов. С другой стороны, частое обновление данных может привести к замедлению работы, поэтому если особой необходимости в этом нет, флажок можно снять.

Установка флажка **Подсвечивать непрочитанные карточки** изменяет отображение папки, если в ней есть непрочитанные карточки: название папки выводится жирным шрифтом, а рядом с названием в скобках приводится количество непрочитанных карточек. Если же флажок снят, отображение названия папки не меняется вне зависимости от присутствия в ней непрочитанных карточек.

В том случае, если в папке находится очень большое количество записей (карточек и ярлыков), их можно просматривать порциями, установив флажок **Применять ограничение количества выводимых записей** и введя количество поочередно выводимых записей в одноименном поле *Количество*.

При желании и для большей наглядности можно сменить значок (иконку) папки, который отображается на левой панели Навигатора; для этого нужно нажать кнопку *Сменить значок* и выбрать иконку из предложенного списка. Восстановить вид значка, определенный по умолчанию, можно при помощи кнопки *Восстановить значок*.

На вкладке *Безопасность* окна свойств стандартной папки определяются права доступа пользователей к папке.

Виртуальные папки предназначены для отображения результатов выполнения расширенного поиска при помощи поискового запроса, указанного в ее свойствах.

Чтобы создать виртуальную папку, следует выделить папку в дереве папок. Диалоговое окно создания новой папки можно открыть несколькими способами:

- выбрать в контекстном меню команду **Создать > Папку**;
- нажать кнопку *Новая папка* панели инструментов Навигатора;
- выбрать команду меню **Файл > Создать > Папка**;
- выбрать команду меню **Файл > Папка > Создать подпапку**.

В открывшемся диалоговом окне вводится название новой папки в соответствующем поле. Устанавливается флажок *Виртуальная* и выбирается из соответствующих раскрывающихся списков *Тип фильтра* и *Фильтр*, по которому в дальнейшем будет производиться отбор (фильтрация) данных в папку.

Для изменения настроек в дереве папок выделяется виртуальная папка, настройки которой необходимо изменить, и открывается диалоговое окно *Свойства папки* одним из следующих способов:

- щелкнуть правой кнопкой мыши по нужной папке и выбрать в появившемся контекстном меню команду *Свойства*;
- нажать кнопку *Свойства* панели инструментов Навигатора;
- выбрать в меню последовательно команды **Файл > Папка > Свойства**.

Диалоговое окно *Свойства папки* виртуальной папки включает четыре вкладки: **Основные свойства**, **Представления**, **Шаблоны** и **Безопасность**; далее уточняются свойства папки, настраиваемые на каждой из вкладок.

На вкладке **Представления** отображается список всех имеющихся представлений. Чтобы разрешить отображение в папке только определенных представлений, устанавливается флажок **Только эти представления могут быть показаны в этой папке** и далее – флажки возле названий представлений, которые могут использоваться для данной папки. Запретить использование любых представлений кроме дайджеста, можно, установив флажок **Только эти представления могут быть показаны в этой папке** и не выбрав ни одного представления.

Чтобы разрешить или запретить определенным пользователям право доступа к какому-либо представлению, необходимо выделить его и нажать кнопку *Доступ*. В открывшемся окне добавить нужного пользователя и определить его права на использование данного представления в текущей папке.

На вкладке **Шаблоны** отображается список всех имеющихся в системе шаблонов. Чтобы разрешить экспорт данных папки только в определенные шаблоны Microsoft Excel, необходимо установить флажок **Только эти шаблоны могут быть использованы с этой папкой**, а затем установить флажки напротив названий шаблонов, которые будут доступны для передачи данных папки в Microsoft Excel.

Вкладка **Основные свойства** позволяет изменить настройки отображения карточек, находящихся в папке, сделав их отличными от установленных по умолчанию или указанных при создании виртуальной папки:

- *Имя папки* – название папки;
- *Тип папки* – виртуальная – информационное поле, не доступное для изменения;
- в поле *Представление* следует указывать представление, которое будет использоваться для папки по умолчанию; в раскрывающийся список этого поля включаются только те представления, которые были разрешены для использования в данной папке на вкладке **Представления**;

- в раскрывающемся списке поля **Шаблон** выбирается для папки название шаблона Excel, который будет использоваться по умолчанию для экспорта данных папки (набор разрешенных для папки шаблонов определяется на вкладке *Шаблоны*).

Следующая группа элементов управления позволяет определить дополнительные настройки свойств папки, служащие для удобства просмотра находящихся в ней карточек.

Флажок **Разрешить использование дайджеста** установлен по умолчанию; его можно снять только в том случае, если на вкладке **Представления** разрешено использование хотя бы одного из существующих представлений. Кнопка **Разрешения** рядом с флажком позволяет назначить определенным пользователям права на использование дайджеста.

Установка флажка **Разрешить автоматическое обновление** приводит к обновлению на экране данных папки через определенный промежуток времени (установленный администратором); в случае снятия флажка обновление данных на экране будет производиться пользователем вручную при помощи кнопки *Обновить* на панели инструментов. С другой стороны, частое обновление данных может привести к замедлению работы, поэтому если особой необходимости в этом нет, флажок можно снять.

Установка флажка **Подсвечивать непрочитанные карточки** изменяет отображение папки, если в ней есть непрочитанные карточки: название папки выводится жирным шрифтом, а рядом с названием в скобках приводится количество непрочитанных карточек. Если же флажок снят, отображение названия папки не меняется вне зависимости от присутствия в ней непрочитанных карточек.

В том случае, если в папке находится очень большое количество записей (карточек и ярлыков), их можно просматривать порциями, установив флажок **Применять ограничение количества выводимых записей** и указав количество поочередно выводимых записей в одноименном поле (*Количество*).

При желании и для большей наглядности можно сменить значок (иконку) папки, который отображается на левой панели Навигатора; для этого необходимо нажать кнопку *Сменить значок* и выбрать иконку из предложенного списка. Восстановить вид значка, определенный по умолчанию, можно при помощи кнопки *Восстановить значок*.

Кнопка *Поиск* позволяет определить новый поисковый запрос для текущей папки или изменить настройки уже существующего: при нажатии на эту кнопку открывается диалоговое окно *Расширенный поиск – Новый запрос*, позволяющее загрузить и отредактировать уже существующий поисковый запрос, или же создать настройки нового запроса и сохранить их под новым именем.

Папка-делегат представляет собой ссылку на одну из стандартных папок иерархии. Доступ к делегату получают пользователи, имеющие доступ к папке, в которой размещен делегат, однако если у сотрудника нет прав доступа к какой-либо карточке, размещенной в делегирующей папке, он не получит доступа к этой карточке и в папке-делегате.

В папке-делегате можно создать новую карточку любого типа, разрешенного для делегирующей папки (карточка появится и в делегирующей папке), но нельзя создать дочернюю папку. Карточка, созданная в папке-делегате, доступна для просмотра и редактирования из делегирующей папки, а также из всех остальных папок-делегатов делегирующей папки.

Папка-делегат наследует некоторые свойства делегирующей папки (папки, на которую она ссылается), а именно:

- название (если папка-делегат размещается в той же папке, что и делегирующая папка, она будет носить то же название, но с номером (2));
- список разрешенных представлений;
- возможность выбора шаблона Excel для экспорта данных представления;
- ссылку на делегирующую папку;

Флажок **Показывать вложенные папки** (по умолчанию флажок не установлен; при установке флажка в папке-делегате будут отображаться все дочерние папки делегирующей папки).

Представление, установленное по умолчанию для отображения содержимого папки-делегата, может отличаться от представления по умолчанию для делегирующей папки, однако может быть выбрано только из списка представлений, разрешенных для использования в делегирующей папке.

Чтобы создать папку-делегат, необходимо выделить правой кнопкой мыши папку, для которой необходимо создать новую папку-делегат и выбрать из контекстного меню команду **Создать делегата на эту папку**. В появившемся окне с иерархией папок выделяется (или создается новая) папка, в которой будет размещена папка-делегат. Далее необходимо нажать кнопку *Выбрать*.

После создания папки-делегата ее свойства можно изменить в диалоговом окне (рис. 5), которое содержит единственную вкладку **Основные свойства**, содержащую следующие поля:

- *Название (имя) папки*;
  - *Тип папки* – Делегат – информационное поле, не доступное для изменения;
  - в поле *Представление* указывается представление, которое будет использоваться для папки по умолчанию; в раскрывающийся список этого поля включаются только те представления, которые были разрешены для использования в делегирующей папке на вкладке *Представления*;
- в раскрывающемся списке поля *Шаблон* выбрать для папки название шаблона Excel, который будет использоваться по умолчанию для экспорта данных папки (набор разрешенных для папки шаблонов определяется на вкладке *Шаблоны*);
- в поле *Связанная папка* указана папка, на которую ссылается папка-делегат: при завершении работы с одной делегирующей папкой нет необходимости создавать новую папку-делегат, можно просто указать другую делегирующую папку.
  - Установка флажка **Показывать вложенные папки** приводит к отображению в папке-делегате всех дочерних папок делегирующей папки.

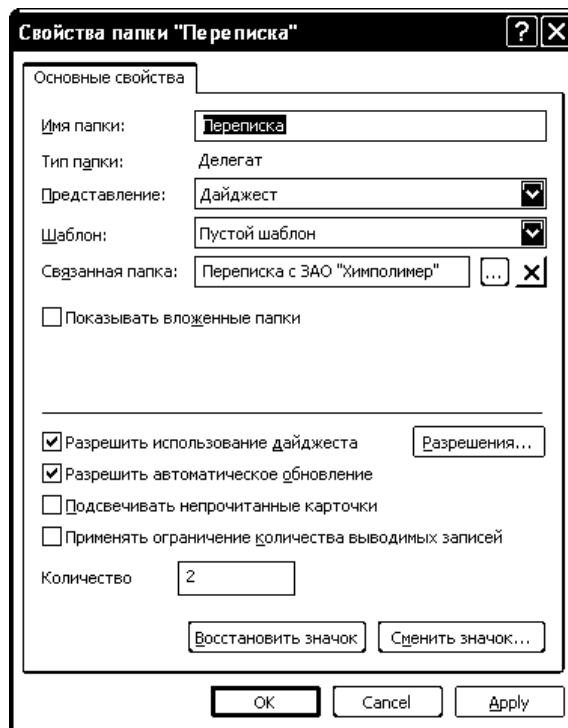


Рис. 5. Диалоговое окно "Свойства папки-делегата"

Следующая группа элементов управления не доступна для изменения *Разрешить использование дайджеста*, кнопка *Разрешения*, *Разрешить автоматическое обновление*, *Подсвечивать непрочитанные карточки*, *Применять ограничение количества выводимых записей* и поле *Количество*; эти настройки идентичны настройкам делегирующей папки.

При желании и для большей наглядности можно сменить значок (иконку) папки, который отображается на левой панели Навигатора; для этого нужно нажать кнопку *Сменить значок* и выбрать иконку из предложенного списка. Восстановить вид значка, определенный по умолчанию, можно при помощи кнопки *Восстановить значок*.

С папками Навигатора можно осуществлять следующие действия:

- Просмотр карточек и ярлыков карточек, находящихся в папке;
- Поиск папки;
- Пометка карточек как прочитанных или непрочитанных;
- Очистка папки от первично удаленных объектов;
- Перемещение папки в определенную папку Навигатора;
- Копирование папки в определенную папку Навигатора;
- Переименование папки;
- Печать содержимого папки;
- Получение URL-адреса папки;
- Экспорт содержимого папки в таблицу Excel;
- Создание папки;
- Удаление папки;
- Определение свойств папки.

## 2.8. РАБОТА С КАРТОЧКАМИ

*Карточка* – элемент системы DocsVision, предназначенный для упорядоченного хранения сведений о внешних объектах и процессах, информация о которых является предметом учета, хранения и обработки. В этом смысле карточка является одним из ключевых элементов системы. Имея, в общем случае, достаточно сложную структуру, каждая карточка является некоторой учетной единицей, позволяя регистрировать одну единицу хранения (например, документ). Карточки создаются пользователем при регистрации документа (или другого объекта, например, файла); чтобы избежать путаницы и беспорядка, желательно сразу создавать карточку в папке, предназначенной для хранения соответствующих документов – определенного типа, относящихся к конкретному подразделению или же в личной папке пользователя.

В общем случае карточка содержит несколько именованных вкладок, каждая из которых несет в себе специфическую информацию о документе или другом объекте, для описания которого создавалась карточка. Каждая вкладка состоит из одной или более секций, в свою очередь представляющих собой набор информационных полей. Не все поля карточек обязательны для заполнения.

Для хранения совместно с регистрационной карточкой документа больших объемов сопутствующей информации к некоторым типам карточек могут прикрепляться файлы, причем в зависимости от планируемого вида дальнейшей обработки как карточки, так и прикрепленных к ней файлов карточка может содержать копию оригинального документа, сам оригинальный документ или ссылку на файл.

После создания карточки можно облегчить доступ к ней при помощи *ярлыка карточки* (аналогичного ярлыкам файлов и программ в Windows): двойной щелчок по ярлыку приводит к открытию карточки. Пользователь может создать ярлык карточки в любой папке, к которой ему разрешен доступ. Левый нижний угол иконки ярлыка отмечен значком стрелки, в отличие от иконки карточки.

При создании множества однотипных документов, содержащих большое количество полей с одинаковой информацией, можно пользоваться шаблонами документов, в которые допускается конвертировать любой уже существующий документ.

*Ярлыком карточки* будем называть ссылку на карточку, которую пользователь в целях удобства обращения к карточке может разместить в любой удобной для него папке.

С помощью ярлыка можно осуществить следующие действия:

- перемещение ярлыка карточки в любую папку Навигатора при помощи команды контекстного меню *Переместить ярлык в папку*;
- копирование ярлыка карточки в любую папку Навигатора с помощью команды контекстного меню *Скопировать ярлык в папку*; при этом все данные карточки дублируются в указанной папке;
- удаление ярлыков карточек, при этом можно удалить как сам ярлык, так и карточку: для выполнения указанных действий предназначены пункты контекстного меню *Удалить карточку* и *Удалить ярлык* соответственно;
- размещение ярлыка некоторых типов карточек в любой папке Навигатора (например, в приложении "Делопроизводство" – при создании карточки процесса или при создании карточки из уже имеющейся карточки).

Ознакомиться со свойствами карточки пользователь может в диалоговом окне *Свойства карточки*, открываемом при выборе из контекстного меню карточки опции *Свойства...* Данное диалоговое окно состоит из четырех вкладок (см. рис. 6): **Общие**, **Ссылки**, **Ярлыки** и **Безопасность**; далее уточним свойства папки, настраиваемые на каждой из вкладок.

На вкладке **Общие** отображаются основные сведения о карточке, сгруппированные по темам в секции:

- дайджест карточки; иконка в левой части этой секции соответствует типу карточки;
- тип карточки;
- размещение карточки в дереве Навигатора;
- даты создания и последнего изменения карточки;
- атрибуты карточки; в этой секции расположен флажок **Архивирована**, установив (сняв) который, пользователь, имеющий соответствующие права, может внести карточку в архив (извлечь карточку из архива);
- URL-адрес карточки.

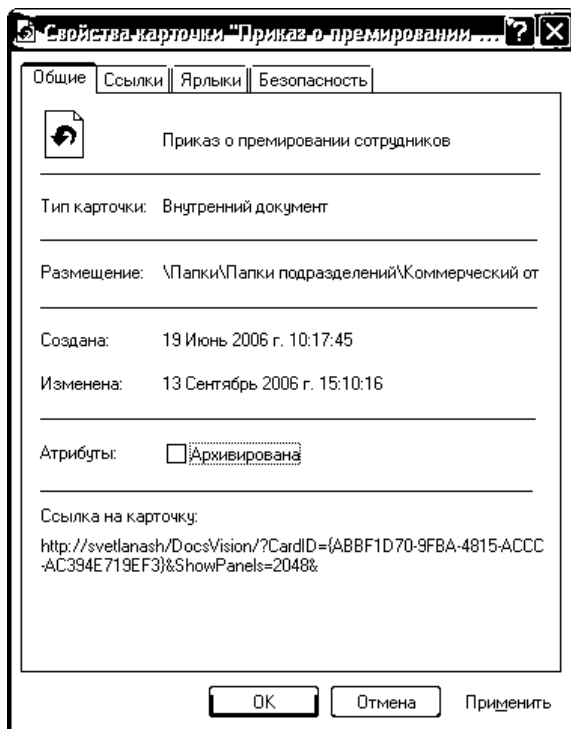


Рис. 6. Диалоговое окно "Свойства карточки"

Основную часть вкладки **Ссылки** составляет список карточек, связанных ссылками с данной. В списке отображаются тип и дайджест карточек; его состав зависит от настроек, указанных в других секциях вкладки.

Переключатель, расположенный на вкладке, может быть установлен в одно из положений:

- **Следующие карточки зависят от данной карточки**; при установленном в этом положении переключателе на вкладке будет отображен список карточек, на которые ссылается данная.
- **Данная карточка зависит от следующих карточек**; при установленном в этом положении переключателе на вкладке будет отображен список карточек, ссылающихся на данную.

Тип ссылки выбирается установкой в секции *Тип ссылки* флажков **Сильная** или **Слабая**, которые могут быть установлены одновременно; при двух снятых флажках список карточек, связанных с данной, останется пустым.

Счетчик *Вложенность* определяет, карточки скольких уровней вложенности войдут в список (например, карточка, непосредственно ссылающаяся на данную, имеет первый уровень вложенности, ссылающаяся на эту ссылку – второй и т.д.). Максимальный предусмотренный для отображения на этой вкладке уровень вложенности – девятый.

На вкладке **Ярлыки** находится список всех ярлыков данной карточки с указанием их типа и расположения в дереве Навигатора. При установленном флажке **Показывать удаленные ярлыки** в список будут включены первично удаленные ярлыки карточки, при этом удаленный ярлык перечеркивается.

На вкладке **Безопасность** окна свойств карточки определяются права доступа пользователей к карточке. Порядок назначения прав доступа к любому объекту происходит одинаково и описывается в соответствующем разделе руководства.

Окно свойств рассматриваемого ниже *шаблона карточки* включают еще одну вкладку – **Шаблоны**, которая будет рассмотрена в соответствующем разделе.

*Шаблон карточки* – это образец карточки определенного типа, в котором заполнена часть полей. Шаблон предназначен для создания большого количества карточек, содержащих сходную информацию, и позволяет сократить время, затрачиваемое на их создание. Шаблоны могут создаваться практически для всех типов карточек.

Карточку, создаваемую при помощи шаблона, в дальнейшем будем называть *экземпляром*. Значения полей экземпляра совпадают со значениями полей шаблона, кроме того, экземпляр содержит копии файлов, прикрепленных к шаблону карточки.

Чтобы создать шаблон по карточке, необходимо выполнить следующие действия:

- щелкнуть правой кнопкой мыши по строке с названием карточки, по которой необходимо создать шаблон;
- выбрать команду контекстного меню **Конвертировать в шаблон**. Иконка карточки будет помечена особым значком DocsVision в правом верхнем углу (например, иконка шаблона карточки файла).

Очевидно, что значения отдельных полей не могут передаваться из шаблона в экземпляр: это поля, значения которых индивидуальны для каждой карточки, например, номера документов карточек приложения "Делопроизводство".

Чтобы создать экземпляр по шаблону, следует выполнить одно из описанных далее действий:

- выделить шаблон на правой панели Навигатора, после чего выбрать из контекстного меню команду **Создать карточку по этому шаблону**;
- дважды щелкнуть в правой панели Навигатора по строке шаблона.

В результате каждого из этих действий будет открыто окно, предлагающее указать папку для размещения экземпляра карточки. После выбора папки вновь созданный экземпляр будет открыт для редактирования.

Разместить экземпляр созданной по имеющемуся шаблону карточки можно в заранее определенной папке. Для задания такой папки необходимо в диалоговом окне *Свойства карточки*, открываемом при выборе из контекстного меню шаблона на карточке опции *Свойства...*, перейти на вкладку **Шаблоны**.

На вкладке **Шаблоны** содержатся:

- поле *По умолчанию создавать карточки в данной папке*, в котором можно указать папку для размещения экземпляров карточки, созданных по данному шаблону; папка выбирается в диалоговом окне *Выбор папки*;
- флажок **Запрашивать папку только при нажатой клавише Shift**; установка флажка приводит к тому, что экземпляр карточки будет создаваться в обозначенной папке при двойном щелчке мышью по карточке шаблона; система будет запрашивать подтверждение выбора папки только в случае нажатой клавиши *Shift*.

При установленном флажке **Запрашивать папку только при нажатой клавише Shift** и пустом поле **По умолчанию создавать** карточки в данной папке экземпляр карточки будет размещен в той же папке, в которой находится шаблон.

При наличии достаточных прав пользователь может создать карточку любого из имеющихся типов (например, входящих в приложение "Делопроизводство" или разработанных организацией самостоятельно). Создание карточки для описания любого объекта (документа, файла, задания и др.) выполняется при помощи одной и той же последовательности действий одним из трех способов:

- с помощью панели инструментов Навигатора или команд контекстного меню;
- из открытой карточки того же типа;
- по шаблону карточки.

Для того чтобы создать карточку из Навигатора: в иерархии папок следует выделить правой кнопкой мыши папку, в которой необходимо создать новую карточку; при этом откроется контекстное меню, в котором можно выбрать:

- команду **Создать**, а затем выбрать тип карточки из списка;
- команды **Создать > Выбрать тип карточки**, и в открывшемся диалоговом окне *Создание новой карточки* выбрать нужный тип карточки.

Существует два режима открытия карточки.

1) Режим *только чтение*, разрешающий только просмотр полей карточки, но не допускающий их редактирования; в этом режиме карточка открывается:

- в области предварительного просмотра;
- при открытии карточки, если у пользователя нет права ее редактирования или если она заблокирована другим пользователем.

2) Режим *редактирования* (обычно установленный по умолчанию), позволяющий пользователю редактировать поля карточки; в данном режиме карточка может находиться только при ее открытии.

Просмотр карточки в режиме только чтения может производиться в области предварительного просмотра на правой панели окна Навигатора. Для отображения/скрытия данной области используется кнопка *Область предварительного просмотра* на панели инструментов Навигатора или команда меню **Вид > Область предварительного просмотра**.

Функция предварительного просмотра может применяться, если в свойствах папки в качестве текущего представления выбран дайджест или если для данного представления разрешен предварительный просмотр.

Для полноценной работы с карточкой необходимо ее открыть в отдельном окне; для этого необходимо дважды щелкнуть левой кнопкой мыши по строке с названием карточки или щелкнуть по ней правой кнопкой мыши и выбрать из контекстного меню команду **Открыть**.

Карточка, открытая одним из пользователей в режиме редактирования, любым другим пользователем может открываться в режиме только чтения, тем самым предотвращается одновременное редактирование ее полей и возможная при этом некорректность сохраняемых данных.

После внесения в карточку необходимых изменений необходимо закрыть карточку щелчком на крестике в ее правом верхнем углу; при этом возможны три варианта: закрыть карточку и сохранить изменения; закрыть карточку и не сохранять изменения или не закрывать карточку и продолжить внесение изменений. Предварительно сохранить карточку можно при помощи кнопки *Сохранить*, расположенной на панели инструментов окна карточки.

Удаление карточек и ярлыков карточек производится в две стадии.

1. *Первичное удаление карточки или ярлыка карточки*. Делает карточку (ярлык) недоступной для просмотра и редактирования; после первичного удаления карточку можно либо восстановить, либо окончательно удалить. Список всех первично удаленных карточек можно просмотреть в папке *Корзина*; список карточек, первично удаленных из какой-либо папки, можно отобразить/скрыть при помощи установки/снятия флажка **Показывать удаленные объекты** в меню **Вид**, при этом строка с записью об удаленной карточке перечеркивается и подсвечивается серым цветом. Для первичного удаления карточки необходимо выделить ее правой кнопкой мыши, а затем выполнить одно из следующих действий:

- выбрать из контекстного меню команду **Удалить**;
- нажать на панели инструментов кнопку **Удалить**;
- нажать на клавиатуре клавишу **<Del>**.

2. *Окончательное удаление карточки*. После окончательного удаления восстановление карточки невозможно. Чтобы окончательно удалить карточку, нужно повторно удалить первично удаленную карточку или выделить не удаленную карточку и воспользоваться сочетанием клавиш **<Shift + Del>**. Освободить папку от всех первично удаленных объектов можно при помощи команды контекстного меню папки **Очистить удаленные объекты**, или контекстного меню системной папки **Корзина Очистить корзину**.

Печать данных карточки, сохранение их в файле формата .xml, а также отправка по электронной почте осуществляется при помощи панели инструментов диалогового окна, активизирующегося при нажатии кнопки панели инструментов карточки *Экспорт и печать*. Все функции, доступные в данном окне (сохранение в файл или помещение в буфер обмена, печать на принтер и отправка по электронной почте), можно применить к:

- шаблону печати;
- файлу формата XML, соответствующему данному бизнес-процессу;
- URL-ссылке на карточку бизнес-процесса.

Выбор осуществляется из раскрывающегося списка поля *Формат*.

На панели инструментов окна *Экспорт и печать* расположены три группы кнопок.

Кнопки первой группы позволяют как непосредственно сохранить данные карточки в выбранном формате (кнопка *Сохранить*), так и скопировать их в буфер для последующей вставки в какой-либо другой файл (кнопка *Копировать в буфер*).

Кнопки второй группы предназначены непосредственно для печати.

Кнопка *Печать* открывает диалоговое окно стандартных настроек печати, позволяющее выбрать принтер для печати, указать страницы, предназначенные для печати, и число копий. Флажок **Печать в файл** может оказаться полезным, если в данный момент печать на принтер по какой-либо причине не может быть осуществлена; при выборе этого флажка по указанному адресу будет сформирован файл формата .rpt, который может быть распечатан в любое время, как только принтер станет доступен. Кроме того, кнопка *Preferences* позволяет настроить дополнительные параметры печати (количество страниц, распечатываемых на одном листе, формат, ориентация и источник бумаги).

Кнопка *Предварительный просмотр* предназначена, как следует из ее названия, для отображения документа в том виде, в каком он будет выводиться на печать, включая колонтитулы и разбиение на страницы. На панели инструментов этого окна находятся кнопки, предназначенные для постраничного просмотра документа, быстрого перемещения на первую и последнюю страницы, и кнопки, позволяющие изменять масштаб просмотра. Кроме того, имеется кнопка настройки печати, позволяющая выбрать в диалоговом окне *Page Setup (Параметры страницы)* размер, ориентацию и источник бумаги, задать отступы от краев бумаги, а также указать надписи, выводимые в колонтитулах документа. Крайняя левая кнопка в соответствии со своим названием *Print (Печать)* вызывает диалоговое окно *Печать* стандартных настроек печати, описанное в предыдущем пункте (вызываемое нажатием кнопки *Печать* панели инструментов диалогового окна *Экспорт и печать*).

Третья и последняя кнопка этой группы приводит к открытию диалогового окна *Page Setup (Параметры страницы)*, описанному в предыдущем пункте.

Последняя группа кнопок панели инструментов представлена кнопками *Отправить почтой* и *Создать ярлык для сотрудника*.

Экспорт данных карточки в файл формата .xml предназначен для сохранения параметров карточки в файловой системе компьютера пользователя.

Для того чтобы экспортировать данные какой-либо карточки в файл .xml необходимо:

- открыть карточку, данные которой предназначены для экспорта;
- на панели инструментов карточки нажать кнопку *Экспорт и печать*;
- в открывшемся окне из раскрывающегося списка поля *Формат* выбрать пункт *XML карточки*;
- нажать на панели инструментов кнопку *Сохранить*, в стандартном диалоговом окне в поле *Название файла (File id)* ввести название файла и нажать кнопку *Сохранить (Save as)*.

Данные карточки в формате .xml могут быть напечатаны на принтере или отправлены по электронной почте. Для этого следует выбрать соответствующие кнопки на панели инструментов диалогового окна *Экспорт и печать* и применить их к формату *XML карточки*.

## 2.9. РАБОТА С ПРЕДСТАВЛЕНИЯМИ

Таблица, в которой отображается информация о карточках в области просмотра содержимого папки, называется представлением. Главным достоинством табличного представления данных является то, что оно позволяет отобразить информацию о карточках различных типов в одной таблице, позволяя выводить на экран только необходимые поля и обеспечивая регулирование высоты строк и ширины столбцов в зависимости от текущих потребностей пользователя.

Для любой папки можно выбрать представление, наиболее наглядно отражающее информацию содержащихся в ней карточек и не включающее избыточной информации.

Поскольку просмотр может осуществляться в целях получения различной информации, можно выбрать для папки несколько представлений, каждое из которых будет отвечать какой-либо одной из поставленных задач (для этого следует создать папку-делегат и определить для нее другое представление).

По умолчанию информация вновь создаваемой папки отображается в виде стандартного представления – дайджеста, кратко описывающего карточку одной строкой текста. Для каждого типа карточки предусмотрен определенный дайджест.

Представление, как и любая таблица, состоит из колонок и строк. Строки таблицы представляют собой карточки и ярлыки карточек. Колонки таблицы выбираются пользователем в зависимости от назначения представления в Настройках представлений данных (например, Автор, Дата создания, Тема).

Пользователь может оставить представление данных, установленное по умолчанию, выбрать его из уже существующих или определить самостоятельно.

По умолчанию устанавливается универсальное представление карточек – дайджест, который может использоваться для карточек любого типа. Дайджест состоит из четырех колонок: *Тип карточки* (входящий документ, исходящий документ, процесс и т.д.), *Дайджест карточки* (для документа – название документа и его номер, для файла – файл и название, для процесса – название), *Создана* (дата создания карточки) и *Изменена* (дата последнего изменения).

Представления, создаваемые пользователем, должны определяться в соответствии с их назначением и включать все необходимые сведения, содержащиеся в карточках (например, для построения отчетов).

## 2.10. ВОЗМОЖНОСТИ ПОИСКА

Платформа DocsVision предоставляет несколько видов поиска документов:

- поиск папки;
- поиск карточки по представлению;
- расширенный поиск;
- поиск карточек при помощи *Поискового агента*.

*Поиск папки* осуществляет контекстный поиск папки по ключевой строке, которая может представлять собой сочетание символов, слово или фразу. В результате такого поиска будет выделена первая найденная папка, отвечающая заданному условию.

*Поиск по представлению* – это контекстный поиск карточек в текущем представлении по ключевой строке, которая может представлять собой сочетание символов, слово или фразу. В результате такого поиска будет выделена первая карточка, отвечающая заданному условию.

*Расширенный поиск* выполняется на основе поисковых запросов, которые позволяют производить поиск карточек в соответствии с определенными критериями. Результаты последнего поиска отображаются в виртуальной папке *Результаты поиска*, которая автоматически создается при первом выполнении расширенного поиска (см. следующий раздел).

Поиск карточек при помощи *Поискового агента* осуществляется на основе запросов, аналогичных запросам для расширенного поиска, однако выполняется системой самостоятельно через определенные интервалы времени и может производиться как при открытом, так и при закрытом основном пользовательском приложении – Навигаторе.

Функция поиска папок осуществляет поиск тех папок, в название которых входит указанная последовательность символов. После указания последовательности символов и направления поиска (вверх или вниз) в дереве будет выделена первая папка, в названии которой найден текст, совпадающий с условием поиска. Поиск производится по всем папкам и подпапкам; если папка, отвечающая условию поиска, является вложенной, ее родительская папка будет развернута, а найденная папка – выделена.

Чтобы осуществить поиск папки, необходимо выполнить следующие действия:

- вызвать окно поиска одним из следующих способов:
  - а) выбрать в главном меню последовательно **Сервис > Найти папку**;
  - б) нажать на панели инструментов Навигатора кнопку *Поиск папки*;
- выбрать из контекстного меню команду *Найти папку*;
- в открывшемся диалоговом окне указать последовательность символов, которая должна присутствовать в названии папки, а также направление поиска *Вверх* или *Вниз*;
- если первая найденная папка не является искомой, необходимо нажать кнопку *Найти далее* для продолжения поиска.

*Поиск по представлению* – это контекстный поиск карточек в текущем представлении по ключевой строке, которая может представлять собой сочетание символов, слово или фразу. В результате такого поиска будет выделена первая найденная карточка, отвечающая заданному условию. Для поиска по представлению может задаваться направление поиска вверх или вниз по списку.

Чтобы осуществить поиск карточки, следует выполнить следующие действия:



- вызвать окно поиска одним из следующих способов:
  - а) выбрать в главном меню последовательно **Сервис > Найти в этом представлении**;
  - б) нажать на панели инструментов Навигатора кнопку *Поиск по представлению*;
- в открывшемся диалоговом окне указывается последовательность символов, которая должна присутствовать в названии карточки, а также направление поиска *Вверх* или *Вниз*;
- если первая найденная карточка не является искомой, необходимо нажать кнопку *Найти далее* для продолжения поиска.

Расширенный поиск карточек выполняется на основе поисковых запросов, позволяющих находить все карточки, отвечающие каким-либо определенным критериям, а также по минимальным имеющимся о них сведениям.

При первом выполнении расширенного поиска автоматически создается папка *Расширенный поиск*, в которой хранятся результаты поиска.

Запросы для расширенного поиска карточек, содержащие описание условий поиска, могут использоваться однократно или сохраняться для последующего использования; при этом пользователь может создать виртуальную папку (или несколько виртуальных папок) для хранения результатов расширенного поиска по какому-либо поисковому запросу (запросам). Обновление результатов поиска будет происходить при каждом обращении к виртуальной папке.

Изменяя свойства виртуальной папки, пользователь имеет возможность строить различные отчеты по данным, отображающимся в области просмотра содержимого папки. Кроме того, можно экспортировать результаты поиска в Microsoft Excel.

Полнотекстовый поиск представляет собой отбор карточек, содержащих определенную строку в тексте карточек документов и/или связанных с документами файлов. Чтобы сделать возможным полнотекстовый поиск по базе данных, на сервере регулярно выполняется индексирование вновь появившихся карточек; при этом новые карточки и файлы будут обнаруживаться при выполнении полнотекстового поиска только после проведения очередного индексирования.

Во время полнотекстового поиска выполняются:

- поиск точного совпадения слов;
- поиск слов, начинающихся с заданной последовательности букв, например, по ключевому слову "приказ" будут отобраны слова "приказываю", "приказать" и т.д.;
- проверка форм слов.

Условия полнотекстового поиска определяются на вкладке *Полнотекстовый* диалогового окна *Расширенный поиск*.

Рассмотрим все доступные в этом окне опции по порядку, в соответствии с секциями, на которые они делятся.

В поле *Найти карточки, в которых встречается строка*, должно находиться слово, несколько слов или фраза, предназначенные для поиска. Если в строку поиска введено несколько слов (разделенных пробелом) или фраза, то будут найдены документы, содержащие все указанные слова. Во время поиска по строке игнорируются слова, не несущие смысловой нагрузки, такие как "и", "and", "1" и т.п.

Область поиска указывается в одноименной секции путем выбора одного из переключателей:

**Искать в тексте карточек и связанных с ними файлов** – поиск осуществляется как в тексте карточек документов, так и в самих документах;

**Искать только в тексте карточек** – поиск осуществляется только в тексте карточек документов;

**Искать только в тексте файлов** – поиск осуществляется только в тексте файлов документов, присоединенных к карточкам;

**Искать файлы по имени, размеру и дате** – поиск осуществляется только по параметрам (имени файла, размеру и дате), определенным при снятом флажке **Искать во всех файлах**.

Если область поиска включает поиск в тексте файлов (в предыдущей секции не выбран флажок **Искать только в тексте карточек**), то можно дополнительно уточнить параметры связанных файлов. Для этого нужно снять флажок **Искать во всех файлах** и задать значения параметров в ставших доступными полях:

- *Маска имени файла*. Для задания поисковой маски используются специальные символы: "\*" – заменяет любое количество произвольных символов, "?" – заменяет один произвольный символ. По умолчанию маска имеет вид \*.\*; при этом поиск выполняется во всех связанных файлах независимо от их имени и расширения.

- *Дата изменения*. Временной интервал для поиска файла по дате внесения в него последних изменений; для этого нужно выбрать в раскрывающемся списке одну из доступных опций (*Любая дата* – осуществляется поиск всех файлов вне зависимости от даты последнего изменения; *Последний день*, *Последних два дня* и др. – осуществляется поиск файлов, в которые были внесены изменения в соответствующий период; при выборе опции *Определено пользователем* в следующих двух полях *С* и *По* – следует задать верхнюю и нижнюю границу временного интервала с помощью календаря).

- *Размер файла*. Условие: больше, меньше или равно заданному размеру файла или любой размер файла.

Атрибутивный поиск представляет собой поиск карточек по значениям их полей. Атрибутивный поиск обладает следующими возможностями:

Поиск может выполняться среди карточек, типы которых указаны на вкладке *Типы* диалогового окна *Расширенный поиск*.

Поиск может выполняться по одному или нескольким полям заданных типов карточек. Условия атрибутивного поиска определяются в соответствии со следующей схемой: <Название поля><Логический оператор><Атрибут: Значение поля/Параметр/Поисковое слово>.

Условия атрибутивного поиска могут объединяться с условиями полнотекстового поиска логическим оператором "ИЛИ" (если достаточно выполнения только одного из условий) или "И" (если необходимо, чтобы выполнялись все условия поиска).

Условия атрибутивного поиска среди карточек одного типа могут объединяться логическим оператором "ИЛИ" (если достаточно выполнения только одного из условий) или "И" (если необходимо, чтобы выполнялись все условия поиска).

## 2.11. ПРАВА ДОСТУПА К ОБЪЕКТАМ

В зависимости от производственной необходимости, а также от обязанностей, исполняемых пользователями в организации, они могут иметь различные права доступа к каждому объекту, в соответствии с которыми и будут осуществлять работу с папками и их содержимым.

Назначение пользователям прав доступа осуществляет администратор (или пользователь с достаточными правами); при этом могут быть назначены права доступа к любым объектам (папкам, карточкам и др.), а также права на создание карточек определенных типов.

При назначении пользователям прав учетные записи могут выбираться как из справочника сотрудников, так и из стандартного справочника учетных записей пользователей.

Открыть вкладку (окно) **Безопасность** можно следующим образом:

- для папок: выбрать из контекстного меню папки команду *Свойства* или выбрать в меню Навигатора команду **Файл > Папка > Свойства**;
- для карточки: выбрать из контекстного меню карточки команду *Безопасность*;
- для справочника: выбрать из контекстного меню справочника команду **Безопасность**;
- для представления: выбрать в меню Навигатора **Вид > Текущее представление > Настройка представлений**, в открывшемся окне *Представления* выделить нужное представление из списка и нажать кнопку *Доступ...* ;
- для поисковых запросов: выбрать на панели инструментов Навигатора кнопку *Расширенный поиск*, в появившемся окне *Расширенный поиск* нажать кнопку *Загрузить*, в открывшемся окне *Загрузка запроса* выделить название запроса и нажать кнопку *Доступ...* .

## 2.12. АРХИВИРОВАНИЕ ОБЪЕКТОВ

Архив Docsvision предназначен для хранения данных, представляющих интерес для организации, но не используемых в ее повседневной деятельности. В Docsvision 3.6 в архив могут быть помещены карточки всех типов, входящих в приложения Делопроизводство и Управление процессами, и карточки тех созданных пользователями типов, для которых предусмотрена архивация.

Архивирование данных является административной задачей, рядовой пользователь не имеет прав на выполнение этой операции. Работать с архивом могут только пользователи, включенные в следующие группы:

*DocsVision Archive Operators*: включенный в эту группу пользователь может архивировать, разархивировать и находить в архиве только те карточки, на работу с которыми у него есть права (минимальные требования – права на чтение);


*DocsVision Administrators*: пользователь, входящий в эту группу, может работать со всеми карточками архива.

Для всех остальных пользователей карточки, помещенные в архив, недоступны и не отображаются в Навигаторе.

Поместить карточку в архив можно двумя способами:

- выполнить следующие действия в Навигаторе:
  - а) выделить карточку в Навигаторе;
  - б) выбрать из контекстного меню опцию *Свойства*;
  - в) на вкладке **Общие** установить флажок **Архивирована**;
- с помощью универсальной функции бизнес-процесса, указав при настройке ее параметров в качестве типа карточку DV, а в качестве функции – *архивировать карточку*. Этим способом можно поместить в архив не только единичную карточку, но и коллекцию.

Для извлечения карточки из архива достаточно снять флажок **Архивирована** на вкладке **Общие** свойств карточки.

Карточки, помещенные в архив, могут отображаться в области просмотра содержимого папки или быть скрыты. Чтобы отобразить (скрыть) отображение архивированных карточек, нужно выбрать в меню Навигатора команду **Вид > Показывать архивированные объекты** (присутствует только у тех пользователей, которым разрешена работа с архивом). Отобразить находящиеся в архиве карточки можно только в дайджесте, но не в пользовательских представлениях; иконка такой карточки в правом нижнем углу помечается буквой А: .

Объект Docsvision относительно архивации может иметь один из четырех статусов:

1. *Не архивирован*: объект не архивирован, с ним можно полноценно работать.
2. *Архивирован*: объект находится в архиве, работа с данными архива запрещена.
3. *Подготовлен к архивации*: объект подготовлен к архивированию. С точки зрения использования, объект уже находится в архиве и работа с его данными запрещена.
4. *Подготовлен к деархивации*: объект подготовлен к возвращению из архива. Работа с его данными все еще запрещена.

Пользователь, которому разрешена работа с архивом, может найти находящуюся там карточку, используя атрибутивный поиск и указав в качестве значения статус архивации *Не архивирован* или *Архивирован* (предварительно необходимо установить флажок *Искать в архивированных карточках* на вкладке полнотекстового поиска).

Пользователи, не имеющие прав на работу с архивом, могут составить запрос на поиск находящихся в нем карточек, но поиск по их запросу не даст результатов.

### Вопросы для самоконтроля

1. Перечислите основные составные части платформы Docsvision.
2. Назовите наиболее важные элементы пользовательского интерфейса Навигатора.

3. Типы папок, доступных пользователю в DocsVision.
4. Типы регистрационных карточек, доступных пользователю в DocsVision.
5. Посредством вызова каких пунктов системного меню осуществляется контроль доступа к объектам DocsVision?

### 3. СЕТИ ПЕТРИ

Важным этапом при внедрении систем электронного документооборота, таких как DocsVision, является моделирование используемых в них бизнес-процессов. Такое моделирование осуществляется посредством асинхронных нотаций (IDEF3, EPC, Activity Diagrams и т.п.). Исследование результирующих моделей на корректность осуществляется средствами математического аппарата сетей Петри, рассмотрению которых посвящена эта глава.

Реальные дискретные системы состоят из разнообразных компонент, различающихся физическими свойствами, функциональным назначением, сложностью внутренней структуры. Для того чтобы сконструировать адекватный математический аппарат, предназначенный для моделирования систем, необходимо установить круг вопросов, которые должны решаться с помощью моделей, и осуществить переход от физических сущностей к их абстракциям, сначала в форме некоторого (ограниченного) набора концептуальных понятий, затем – в точных математических терминах.

Первый шаг на пути к построению модели дискретной системы – это абстрагирование от конкретных физических и функциональных особенностей ее компонент. Компоненты системы и их действия представляют абстрактными *событиями*, каковыми могут быть, например, исполнение оператора программы, переход триггера из состояния в состояние, прерывание в операционной системе, операция станка или конвейерной линии, завершение этапа проекта и т.п.

Событие может произойти (реализоваться) один раз, повториться многократно, порождая конкретные *действия* (реализация события), или не произойти ни разу. Совокупность действий, возникающих как реализации событий при функционировании дискретной системы, образуют *процесс*, порождаемый этой системой.

Реальная система функционирует во времени, события происходят в некоторые моменты времени и длятся некоторое время. В синхронных моделях дискретных систем события явно привязаны к определенным моментам или интервалам времени, в которые происходит одновременное изменение всех состояний всех компонент системы, трактуемое как изменение общего состояния системы. Смена состояний происходит последовательно. Этот подход к моделированию больших параллельных систем имеет ряд недостатков.

Во-первых, в большой системе приходится учитывать состояние всех компонент при каждой смене ее общего состояния, что делает модель громоздкой, особенно в тех случаях, когда локальные изменения касаются небольшого фрагмента системы.

Во-вторых, при таком подходе исчезает информация о причинно-следственных связях между событиями в системе. Например, если два события при функционировании системы произошли одновременно, то мы не знаем, произошло ли это случайно или в этом факте скрыт какой-то функциональный смысл. Такие понятия, как конфликты между компонентами системы (из-за ресурсов) или ожидание одним из компонентов результатов работы других компонентов, трудно выражаются в терминах смены состояний системы.

В-третьих, в так называемых асинхронных системах события могут происходить внутри неопределенно больших интервалов времени, заранее трудно или нельзя указать более точно время их начала, конца и длительность.

Выходом может служить отказ от введения в модели дискретных систем времени и тактовых последовательностей изменений состояний, а замена их – причинно-следственными связями между событиями. Модели такого типа (в том числе сети Петри) называют *асинхронными*. (Если возникает необходимость осуществить привязку ко времени, то моменты или интервалы времени представляют как события. Таким образом, существенно синхронные системы могут описываться в терминах асинхронных моделей.) Замена временных связей причинно-следственными дает возможность более наглядно описать структурные особенности функционирования систем.

Отказ от времени приводит к тому, что события в асинхронной модели рассматриваются или как элементарные (неделимые, "мгновенные"), или как составные, имеющие некоторую внутреннюю структуру, образованную из "подсобытий". При неформальном описании функционирования асинхронных моделей, в частности, сетей Петри, мы будем привлекать временные отношения (раньше, позже, одновременно и т.п.), когда это удобно или привычно, но они представляют лишь результаты причинно-следственных отношений.

Взаимодействие событий в больших асинхронных системах имеет, как правило, сложную динамическую структуру. Эти взаимодействия описываются более просто, если указывать не непосредственные связи между событиями, а те ситуации, при которых данное событие может реализоваться. При этом глобальные ситуации в системе формируются с помощью локальных операций, называемых *условиями* реализации событий.

Условие имеет емкость: условие не выполнено (емкость равна 0), условие выполнено (емкость равна 1), условие выполнено с  $n$ -кратным запасом (емкость равна  $n$ ,  $n$ -целое положительное число). Условие соответствует таким ситуациям в моделируемой системе, как наличие данного для операции в программе, состояние некоторого регистра в устройстве ЭВМ, наличие деталей на конвейере и т.п. Определенные сочетания условий разрешают реализоваться некоторому событию (предусловия события), а реализация события изменяет некоторые условия (постусловия события), т.е. события взаимодействуют с условиями, а условия с событиями.

Таким образом предполагается, что для решения указанных в начале главы задач достаточно представлять дискретные системы как структуры, образованные из элементов двух типов – событий и условий.

В сетях Петри события и условия представлены абстрактными символами из двух непересекающихся алфавитов, называемых соответственно множеством *переходов* и множеством *мест*. В графическом представлении сетей переходы изображаются "барьерами", а места – кружками (см. рис. 7 – 9). Условия-места и события-переходы связаны отношением непосредственной зависимости (непосредственной причинно-следственной связи), которое изображается с помощью дуг, ведущих из

мест в переходы и из переходов в места. Места, из которых ведут дуги на данный переход, называются его *входными* местами. Места, на которые ведут дуги из данного перехода, называются его *выходными* местами.

В сети на рис. 7, а места  $p_1$  и  $p_2$  являются входными для перехода  $t_1$ , а места  $p_3$  и  $p_4$  – выходными. В этом примере событие-переход  $t_1$  непосредственно зависит от условий-мест  $p_1$  и  $p_2$ , а места  $p_3$  и  $p_4$  непосредственно зависят от  $t_1$ . В этой же сети место  $p_2$  является входным одновременно для двух переходов  $t_1$  и  $t_2$ , место  $p_6$  является выходным одновременно для двух переходов  $t_3$  и  $t_4$ .

Выполнение условия изображается разметкой соответствующего места, а именно помещением числа  $n$  или  $n$  фишек (маркеров) в это место, где  $n > 0$  – емкость условия:

- – условие  $p$  не выполнено;
- – условие  $p$  выполнено;
- ⊙ – условие  $p$  имеет емкость 3;
- ⊕ – условие  $p$  имеет емкость 5.

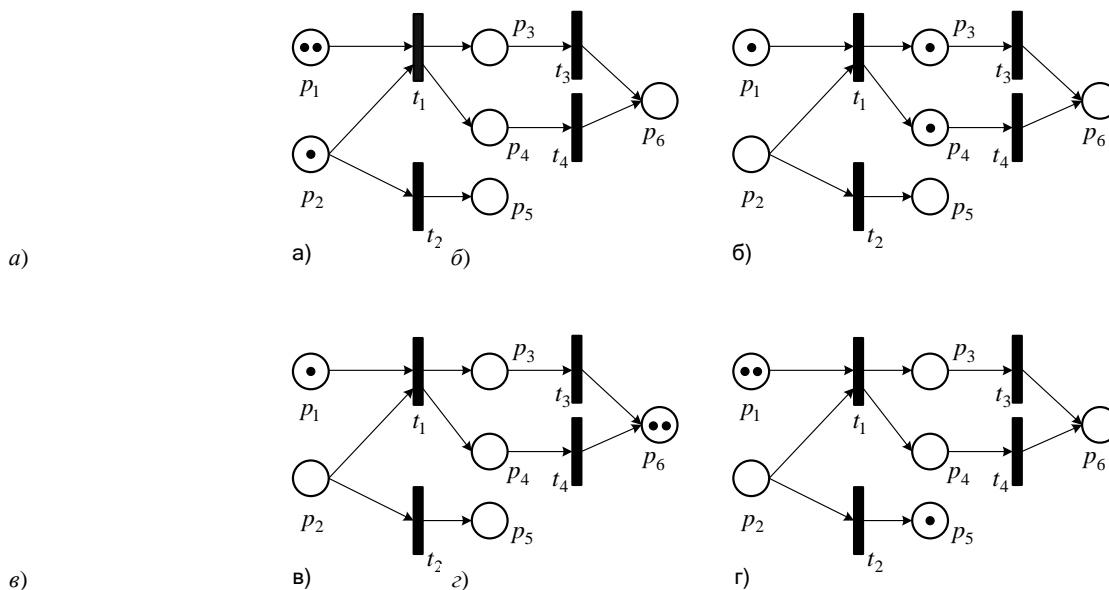


Рис. 7. Пример 1

Динамика поведения моделируемой системы находит свое отражение в функционировании (работе) сети Петри. Неформально работу сети можно представить как совокупность локальных действий, которые называются *срабатываниями* переходов. Они соответствуют реализациям событий и приводят к изменению разметки мест, т.е. к локальному изменению условий в системе.

Переход может сработать если выполнены все условия реализации соответствующего события. Например, для так называемых ординарных сетей Петри (частный случай принятой в настоящее время версии сетей Петри, введенный им в первой работе) все входные места перехода должны содержать хотя бы по одной фишке.

Срабатывание перехода – неделимое действие, изменяющее разметку его входных и выходных мест следующим образом: из каждого входного места изымается по одной фишке. Тем самым реализация события, изображаемого переходом, изменяет состояние (емкость) непосредственно связанных с ним условий так, что емкость предусловий, вызвавших реализацию этого события, уменьшается, а емкость постусловий, на которые оно влияет, увеличивается. Переход  $t_1$  на рис. 7, а может сработать, так как оба его входных места  $p_1$  и  $p_2$  содержат фишки, а после срабатывания  $t_1$  разметка его входных и выходных мест изменяется так, как показано на рис. 7, б.

Если два (и более) перехода могут сработать и они не имеют общих входных мест, то их срабатывания являются независимыми действиями, осуществляемыми в любой последовательности или параллельно.

Если несколько переходов могут сработать и имеют общее входное место (как переходы  $t_1$  и  $t_2$  на рис. 7, а), то срабатывает только один, любой из них. При этом может оказаться, что, сработав, этот переход лишит возможности сработать другие переходы (рис. 7, б и в). Таким способом события могут исключить возможность реализации других. В сети никак не указывается, каким образом конфликт следует фактически разрешить. Считается, что решение о том, какое из конфликтующих событий следует реализовать, принимается вне формализма сети, т.е. поведение сети носит недоопределенный недетерминированный характер. Аналогичный конфликт возникает в том случае, когда несколько переходов могут сработать и они имеют общие выходные места, как переходы  $t_3$  и  $t_4$  (рис. 7, б и в).

В процессе функционирования сети происходит смена разметок мест как результат срабатывания ее переходов. Сеть останавливается, если ни один из ее переходов не может сработать (рис. 7, в и г).

На рис. 8, а показан еще один пример сети Петри с некоторой начальной разметкой мест, при которой может сработать только переход  $t_1$ , так как его единственное входное место  $p_1$  содержит фишку. Переходы  $t_2$ ,  $t_5$ , и  $t_6$  имеют по одному входному месту, не содержащему фишки, поэтому они не могут сработать. Переходы  $t_3$  и  $t_4$  имеют по два входных места. Это означает, что общее условие реализации события, представленного переходом  $t_3$  или  $t_4$ , является конъюнкцией из двух условий. Для каждого из переходов-событий выполнено лишь одно условие, поэтому ни  $t_3$ , ни  $t_4$  не могут сработать.

В результате срабатывания перехода  $t_1$  место  $p_1$  лишится фишки, а места  $p_3$  и  $p_5$  получают по одной фишке (рис. 8, б). Изменилась разметка сети и при новой разметке могут сработать два перехода –  $t_3$  и  $t_4$ . Срабатывание любого из них помещает фишку в место  $p_7$ , после чего возможно срабатывание переходов  $t_5$  и  $t_6$ , то место  $p_7$  будет содержать две фишки (рис. 8, в), в противном случае – одну. Место  $p_7$  – общее входное место для переходов  $t_5$  и  $t_6$ . Если  $p_7$  содержит одну фишку, то сможет сработать только один из переходов  $t_5$  или  $t_6$ , так как сработавший переход заберет единственную фишку. Если же  $p_7$  содержит две фишки, то возможны различные продолжения работы сети:

- 1) сработает переход  $t_5$ , затем – переход  $t_6$ ;
- 2) сработает переход  $t_6$ , затем – переход  $t_5$ ;
- 3) дважды сработает  $t_5$ , а  $t_6$  не может реализовать возможность срабатывания;
- 4) дважды сработает  $t_6$ , а  $t_5$  не сможет сработать ни разу.

В первых двух случаях места  $p_1$  и  $p_2$  получают по фишке, в последних двух случаях одно из мест  $p_1$  или  $p_2$  будет содержать две фишки, а второе не будет иметь фишек. Если  $p_1$  и  $p_2$  содержат по фишке, то срабатывание переходов  $t_1$  и  $t_2$  приведет к разметке, которая уже возникала ранее в процессе функционирования сети (рис. 8, б) и последующая работа сети будет повторять описанную выше. Если же одно место содержит две фишки (например,  $p_2$ ), то сработать может (дважды) только один переход (в данном случае  $t_2$ ), после чего каждый из переходов  $t_3$  и  $t_4$  имеет в одном входном месте две фишки, а в другом – ни одной (рис. 8, г). Возникает разметка, при которой ни один из переходов сети не может сработать, и сеть Петри останавливается.

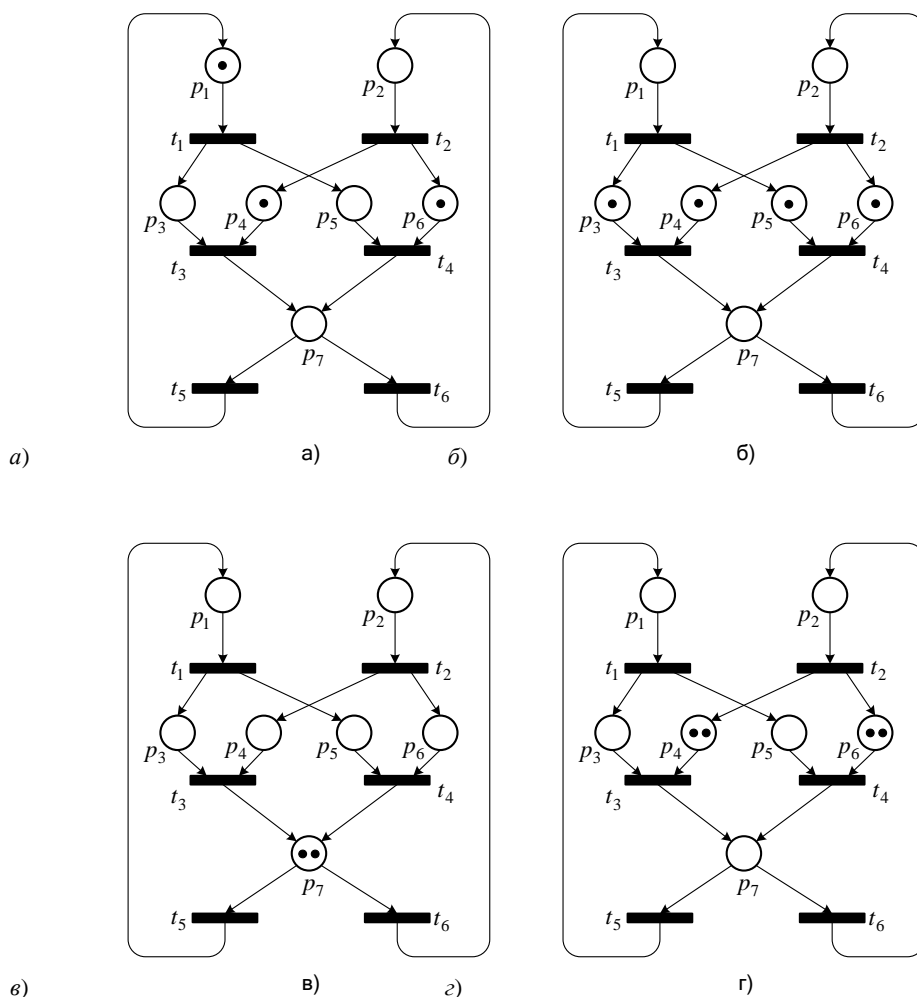


Рис. 8. Пример 2

Анализ работы сети Петри, показанный на рис. 8, позволяет сделать некоторые выводы о функционировании моделируемой сетью системы (это может быть, например, фрагмент операционной системы, состоящей из параллельных циклических процессов, взаимодействующих друг с другом). В частности, можно отметить, что система способна функционировать циклически как угодно долго, но может и остановиться в "тупиковой" ситуации, показанной на рис. 8, 2.

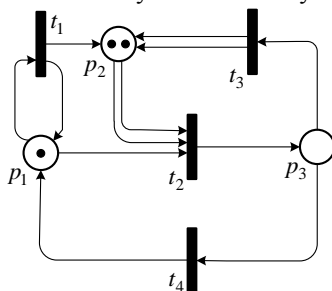


Рис. 9. Пример 3

Таким образом, сети Петри формализуют понятие абстрактной асинхронной системы – динамической структуры из событий и условий. В общей теории сетей сеть Петри рассматривается как один из способов сетевого моделирования систем. Вводятся более общие сетевые модели. Их единую основу образует понятие неинтерпретированной ориентированной сети из условий и событий, которая описывает только статическое строение системы (формальное определение сети дано дальше). Самой общей в спектре динамических сетевых моделей является, по-видимому, условно-событийная система, которая представляет собой сеть, дополненную правилами изменений условий в результате реализации событий. Сеть Петри можно считать конкретизацией условно-событийной системы.

Если сеть Петри описывает функциональную схему моделируемой системы, то работа сети моделирует процесс, происходящий при функционировании системы. Поскольку процесс протекает во времени, для его изучения нужно зафиксировать его в форме некоторой "истории процесса", которую обычно отождествляют с самим процессом. Недетерминированный характер функционирования асинхронной системы и соответствующей сети Петри приводит к тому, что система может породить не единственный процесс, а множество возможных процессов. Кроме того, процессы, порождаемые такими системами, являются параллельными.

Возникает вопрос, каким образом формализовать понятие параллельного процесса, в какой системе понятий можно удобно и полно описывать параллельные процессы (а также множества параллельных процессов) и изучать их. Другими словами, возникает необходимость в разработке моделей параллельных процессов. Поскольку параллельный процесс можно рассматривать как дискретную динамическую систему, то в этом случае можно использовать сетевую модель, которая является частным случаем условно-событийной системы. Модели такого типа будем называть реализационными сетями или сетями-процессами. Они представляют собой сети, в общем случае бесконечные, с дополнительными ограничениями на структуру связей между условиями и событиями и на начальную разметку условий.

Возможность описывать системы и порождаемые ими процессы в рамках одного и того же формализма сетей позволяет не только унифицировать математический аппарат теории систем и процессов, но и более наглядно выявлять связи между функциональными и операционными свойствами систем.

### 3.1. ФОРМАЛИЗАЦИЯ

Сеть Петри – это набор  $N = (P, T, F, W, M_0)$ , где  $(P, T, F)$  – конечная сеть (множество  $X = P \cup T$  конечно), а  $W : F \rightarrow n \setminus \{0\}$  и  $M_0 : P \rightarrow N$  – две функции, называемые соответственно *кратностью дуг* и *начальной разметкой*. Первая сопоставляет каждой дуге число  $n > 0$  (кратность дуги). Если  $n > 1$ , то в графическом представлении число  $n$  выписывается рядом с короткой чертой, пересекающей дугу. Часто такая дуга будет также заменяться пучком из  $n$  дуг, соединяющих соответствующие элементы сети. Условимся никак не отмечать кратность дуг, равную 1. Такую сеть будем называть *ординарной*. Вторая функция сопоставляет каждому месту  $p \in P$  некоторое число  $M_0(p) \in N$  (разметка места). В графическом представлении сети разметка места  $p$  изображается помещением в вершину-кружок числа  $M_0(p)$  или, если это число невелико, соответствующего числа точек (фишек).

Функционирование сети Петри описывается формально с помощью множества последовательностей срабатываний и множества достижимых в сети разметок. Эти понятия определяются через правила срабатывания переходов сети.

Разметка сети  $N$  – это функция  $M : P \rightarrow N$ . Если предположить, что все места сети  $N$  строго упорядочены каким-либо образом, т.е.  $P = (p_1, \dots, p_n)$ , то разметку  $M$  сети (в том числе начальную разметку) можно задать как вектор чисел  $M = (m_1, \dots, m_n)$  такой, что для любого  $i$ ,  $1 \leq i \leq n$ ,  $m_i = M(p_i)$ . Если  $P' = \{p_{i_1}, \dots, p_{i_k}\}$  – подмножество мест из  $P$ , то условимся через  $M(P')$  обозначать множество разметок  $\{M(p_{i_1}), \dots, M(p_{i_k})\}$ . Если  $P'$  представить как вектор  $P' = (p_{i_1}, \dots, p_{i_k})$ , то  $M(P')$  обозначает вектор из множества  $N^k$ , называемый *проекцией* разметки  $M$  на  $P'$ . На основе отношения инцидентности  $F$  и функции кратности дуг  $W$  можно ввести *функцию инцидентности*  $F : P \times T \cup T \times P \rightarrow N$ , которая определяется как

$$F(x, y) = \begin{cases} n, & \text{если } xFy \wedge (W(x, y) = n), \\ 0, & \text{если } \neg(xFy). \end{cases}$$

Если места сети упорядочены, то можно каждому переходу  $t$  сопоставить два целочисленных вектора  $\bullet F(t)$  и  $F \bullet(t)$  длиной  $n$ , где  $n = |P|$ :

$$\begin{aligned} \bullet F(t) &= (b_1, \dots, b_n), \text{ где } b_i = F(p_i, t); \\ F \bullet(t) &= (b_1, \dots, b_n), \text{ где } b_i = F(t, p_i). \end{aligned}$$

Переход  $t$  может сработать при некоторой разметке  $M$  сети  $N$ , если  $\forall p \in \bullet t : M(p) \geq F(p, t)$ , т.е. каждое входное место  $p$  и  $t$ . Это условие можно переписать в векторной форме следующим образом:

$$M \geq \bullet F(t).$$

Для ординарной сети Петри условие срабатывания перехода означает, что любое входное место этого перехода содержит хотя бы одну фишку, т.е. ненулевую разметку.

Срабатывание перехода  $t$  при разметке  $M$  порождает разметку  $M'$  по следующему правилу:

$$\forall p \in P : M'(p) = M(p) - F(p, t) + F(t, p),$$

т.е.

$$M' = M - \bullet F(t) + F \bullet(t).$$

Таким образом, срабатывание перехода  $t$  изменяет разметку так, что разметка каждого его входного места  $p$  уменьшается на  $F(p, t)$ , т.е. на кратность дуги, соединяющей  $p$  и  $t$ , а разметка каждого его выходного места увеличивается на  $F(t, p)$ , т.е. на кратность дуги, соединяющей  $t$  и  $p$ .

На множестве разметок можно ввести отношение  $[ \rangle$  непосредственного следования разметок:

$$M [ \rangle M' \Leftrightarrow \exists t \in T : (M \geq \bullet F(t)) \wedge (M' = M - \bullet F(t) + F \bullet(t)).$$

Будем использовать уточняющее обозначение  $M [ \rangle M'$ , если  $M'$  непосредственно следует после  $M$  в результате срабатывания перехода  $t$ .

Говорят, что разметка  $M'$  достижима от разметки  $M$ , если существует последовательность разметок  $M, M_1, M_2, \dots, M'$  и слово  $\tau = t_1 t_2 \dots t_k$  в алфавите  $T$  такое, что

$$M [ t_1 \rangle M_1 [ t_2 \rangle M_2 \dots [ t_k \rangle M'.$$

Слово  $\tau$  в этом случае называется последовательностью срабатываний, ведущих от  $M$  к  $M'$ . Обобщим отношения непосредственного следования до отношения  $M'$  достижима от  $M$ , используя обозначение  $M [ \rangle M'$  или  $M [\tau \rangle M'$ , если уточняется последовательность срабатываний. (Последовательность  $\tau$  может быть пустой, т.е.  $M$  достижима от  $M$ .)

Множество  $\{M' \mid M [ \rangle M'\}$  разметок, достижимых в сети  $N$  от разметки  $M$ , обозначим через  $R(N, M)$ . Множество  $R(N) = R(N, M_0)$ , т.е. множество всех разметок, достижимых в  $N$  от начальной разметки  $M_0$ , называют множеством *достижимых разметок сети  $N$* . (Заметим, что  $M \in R(N, M)$  и  $M_0 \in R(N)$ .)

Множеством последовательностей срабатываний сети  $N$ , или свободным языком сети  $N$ , называется множество

$$L(N) = \{\tau \in T^* \mid \exists M \in R(N) : M_0 [\tau \rangle M\},$$

т.е. множество всех последовательностей срабатываний, ведущих от  $M_0$  к каждой достижимой в  $N$  разметке.

На рис. 9 изображена сеть Петри, на примере которой поясним данные выше определения. В этой сети  $P = \{p_1, p_2, p_3\}$ ,  $T = \{t_1, t_2, t_3, t_4\}$ . Функция инцидентности  $F$  задается с помощью следующих двух таблиц, в которых на пересечении строки  $x$  и  $y$  стоит число  $F(x, y)$ :

	$p_1$	$p_2$	$p_3$
$t_2$	1	1	0
$t_2$	0	0	1
$t_3$	0	2	0
$t_4$	1	0	0

	$t_2$	$t_2$	$t_3$	$t_4$
$p_1$	1	1	0	0
$p_2$	0	2	0	0
$p_3$	0	0	1	1

Начальная разметка  $M_0$  задается следующим образом:  $M_0(p_1)=1$ ,  $M_0(p_2)=2$ ,  $M_0(p_3)=0$  или в векторной форме:  $M_0=(1, 2, 0)$ .

При разметке  $M_0$  могут работать переходы  $t_1$  и  $t_2$ , так как  $M_0=(1, 2, 0) \geq \bullet F(t_1)=(1, 0, 0)$ ,  $M_0 \geq \bullet F(t_2)=(1, 2, 0)$ . Переходы  $t_3$  и  $t_4$  не могут работать, так как  $M_0 \geq F(t_3)=(0, 0, 1)$ ,  $M_0$ .

В результате срабатывания перехода  $t_1$  разметка  $M_0$  сменяется на разметку  $(1, 3, 0)$ , а в результате срабатывания перехода  $t_2$  разметка  $M_0$  сменяется на разметку  $(0, 0, 1)$ . Обе новые разметки непосредственно следуют после  $M_0$  в рассматриваемой сети  $N$ , происходящие в результате срабатывания ее переходов, в виде *графа разметок* – ориентированного графа, множество вершин которого образовано множеством  $R(N)$  достижимых в  $N$  разметок. Из вершины  $M$  в вершину  $M'$  ведет дуга, помеченная символом перехода  $t$ , если и только если  $M[t]M'$ .

Разметка  $M \in R(N)$  называется тупиковой. Если в сети  $N$  не существует ни одного перехода, который может работать при этой разметке. Для рассматриваемой сети тупиковыми являются разметки  $(0, 2, 0)$ ,  $(0, 3, 0)$ ,  $(0, 4, 0)$ , ...,  $(0, n, 0)$ , ...

Легко видеть, что если выделить путь по дугам графа разметок, начинающийся в вершине  $M$  и заканчивающийся в вершине  $M'$ , и выписать подряд все встречающиеся символы переходов, то полученное слово образует последовательность срабатываний, ведущих от  $M$  к  $M'$ . Множество всех слов, полученных выписыванием символов переходов вдоль путей, начинающихся в  $M_0$ , образует множество последовательностей срабатываний сети, или ее свободный язык. Так, язык рассматриваемой сети включает слова

$$\{\lambda, t_1, t_2, t_1t_1, t_1t_2, t_2t_3, t_2t_4, t_1t_1t_1, t_1t_2t_1, t_1t_2t_3, t_1t_2t_4, t_2t_4t_1, t_1t_1t_1t_1, t_1t_1t_1t_2, t_1t_1t_2t_3, t_1t_1t_2t_4, t_1t_2t_4t_1, \dots\}.$$

Следующая теорема характеризует эффект увеличения начальной разметки в сети.

**Теорема.** Пусть  $N=(P, T, F, W, M_0)$ ,  $N'=(P, T, F, W, M_0+K)$ , где  $K \in N^{|P|}$ . Тогда

а)  $M \in R(N) \Rightarrow (M+K) \in R(N')$ ;

б)  $L(N) \subseteq L(N')$ ;

в)  $M_1[\tau]M_2 \Rightarrow (M_1+K)[\tau](M_2+K)$ .

**Доказательство.** Воспользуемся индукцией по длине срабатываемых последовательностей. Прежде всего  $M_0 \in R(N)$  и  $(M_0+K) \in R(N')$ . Заметим также, что  $M_0[\lambda]M_0$ , где  $\lambda$  – пустая последовательность,  $\lambda \in L(N)$  и  $\lambda \in L(N')$ . Предположим, что

$$M_0[\tau]M \text{ в } N \text{ и } (M_0+K)[\tau](M+K).$$

Покажем, что

$$\forall t \in T: M[t]M' \Rightarrow (M+K)[t](M'+K).$$

Если  $t$  может работать при разметке  $M$  в сети  $N$ , т.е.  $M \geq \bullet F(t)$ , то  $t$  может работать и при разметке  $(M+K) \geq M$  в сети  $N'$ . Это означает, что последовательность срабатываний  $\tau t$  принадлежит как  $L(N)$ , так и  $L(N')$ . (Если  $t$  не может работать при  $M$  в  $N$ , то это не исключает возможности срабатывания  $t$  при  $(M+K) \geq M$  в сети  $N'$ .) После срабатывания перехода  $t$  в  $N$  и соответственно в  $N'$  разметка в обеих сетях изменяется следующим образом:

$$M - \bullet F(t) + F \bullet(t) = M';$$

$$(M+K) - \bullet F(t) + F \bullet(t) = (M - \bullet F(t) + F \bullet(t)) + K = M' + K.$$

Тем самым установлена справедливость соотношений а) и б). Аналогично можно убедиться в том, что имеет место соотношение в).  $\square$

### 3.2. СВОЙСТВА СЕТЕЙ ПЕТРИ И ИХ АНАЛИЗ

Для постановок задач анализа сетей Петри прежде всего необходимо указать и формально определить те свойства сетей, которые целесообразно анализировать. Естественно, что при выборе таких свойств главную роль играет ориентация на практические задачи, возникающие при исследовании моделируемых сетями дискретных систем. Когда такие свойства выделены, становится вопрос о возможности их автоматического распознавания в произвольных сетях Петри или в некоторых частных случаях сетей. Наконец, третий этап исследований состоит в нахождении оптимальных алгоритмов распознавания свойств сетей, которые оказываются принципиально распознаваемыми.

#### 3.2.1. Основные свойства сетей Петри

Первое из рассматриваемых ниже свойств сетей Петри связано с ограниченной емкостью реальных условий реализации событий. Действительно, из определения правил срабатывания переходов сети следует, что для реализации события, моделируемого некоторым переходом, достаточно, чтобы каждое его входное место-условие содержало некоторое конечное чис-



ло фишек, равное кратности дуги, соединяющей его с переходом. Однако при работе сети Петри общего вида некоторые ее места могут накапливать неограниченное число фишек (например, место  $p_2$  в сети на рис. 9). Если интерпретировать места как накопители (буферы) данных, сигналов или деталей в моделируемых системах, то естественно потребовать. Что при любом варианте функционирования этих систем не происходило бы переполнение накопителей, которые в реальных ситуациях имеют конечную, фиксированную емкость. Следующие понятия формализуют такие требования.

Место  $p$  в сети Петри  $N = (P, T, F, W, M_0)$  называется *ограниченным*, если существует число  $n$  такое, что для любой достижимой в сети разметки  $M$  справедливо неравенство  $M(p) \leq n$ . Сеть  $N$  называется *ограниченной сетью*, если любое ее место ограничено. Ясно, что множество достижимых разметок  $R(N)$  конечно, если и только если  $N$  – ограниченная сеть. В сети на рис. 9 места  $p_1$  и  $p_3$  ограничены, так как каждое из них может содержать не более одной фишки. В то же время место  $p_2$  не ограничено, и поэтому эта сеть не является ограниченной. Сеть на рис. 8 ограничена: любое место в ней не может содержать более двух фишек. Место  $p$  называется *безопасным*, если  $\forall M \in R(N) : M(p) \leq 1$ ; соответственно сеть безопасна, если все ее места безопасны. Любая достижимая в безопасной сети разметка представляет собой вектор из 0 и 1. Сети на рис. 8 и рис. 9 не являются безопасными. Ограниченность и безопасность характеризует емкость условий: в дискретной информационной системе, моделируемой соответствующими сетями, можно ограничить емкость накопителей, необходимых для хранения условий наступления событий. Родственным этим понятиям является понятие консервативной сети, в которой сумма фишек во всех ее местах остается постоянной при работе сети, т.е.

$$\forall M_1, M_2 \in R(N) : \sum_{p \in P} M_1(p) = \sum_{p \in P} M_2(p).$$

В консервативной сети каждый переход консервативен в том смысле, что его срабатывание не меняет число фишек в сети, т.е.  $|\bullet t| = |t \bullet|$ .

Переход в сети может сработать при определенных условиях, связанных с разметкой его входных мест. В общем случае может оказаться, что для некоторого перехода условие его срабатывания никогда не выполняется, как бы ни функционировала сеть. Такой переход – лишний в сети, его можно исключить без ущерба для работы сети. Может случиться также, что после некоторой последовательности срабатываний переходов сети и соответствующих изменений ее разметки некоторые переходы, в том числе те, которые уже срабатывали, больше никогда не сработают, какие бы варианты достижимых в сети разметок не возникали. Это означает, что в моделируемых системах могут появляться ситуации, тупиковые для некоторых событий, "исключающие их из дальнейшей игры". Например, в операционных системах подобные случаи имеют место при взаимных блокировках процессов (deadlocks). Следующие определения связаны с выявлением таких ситуаций в сетях Петри.

Переход  $t$  в сети Петри  $N = (P, T, F, W, M_0)$  называется *потенциально живым при разметке*  $M \in R(N)$ , если

$$\exists M' \in R(N, M) : M' \geq \bullet F(t),$$

т.е. существует достижимая от  $M$  разметка  $M'$ , при которой переход  $t$  может сработать. Если  $M = M_0$ , то  $t$  называется *потенциально живым в сети*  $N$ . Переход  $t$  – *мертвый при*  $M$ , если он не является потенциально живым при  $M$ . Переход  $t$  – *мертвый*, если он мертв при любой достижимой в сети разметке.

Переход  $t$  в сети Петри  $N$  называется *живым*, если

$$\forall M \in R(N), \exists M' \in R(N, M) : M' \geq \bullet F(t),$$

т.е. переход  $t$  является потенциально живым при любой достижимой в сети  $N$  разметке. Переход  $t$  – *потенциально мертвый*, если существует  $M \in R(N)$  такая, что при любой разметке  $M' \in R(N, M)$  переход  $t$  не может сработать. Разметка  $M$  в этом случае называется  $t$  – *тупиковой*; если она  $t$  – тупиковая для всех  $t \in T$ , то она является тупиковой. Если  $M$  – тупиковая разметка, то  $R(N, M) = \{M\}$ . Сеть называется *живой*, если все ее переходы живы.

В сети на рис. 8 все переходы потенциально живы и все переходы потенциально мертвы, так как в ней достижима тупиковая разметка (рис. 8, 2). Эта сеть не может быть живой, так как содержит потенциально мертвые переходы. Сеть на рис. 9 также не является живой, так как в ней достижимы тупиковые разметки вида  $(0, n, 0)$ ,  $n \geq 0$ .

Переход  $t$  называется *устойчивым* в сети  $N$ , если

$$\forall t' \in T \setminus \{t\}, \forall M \in R(N);$$

$$(M \geq \bullet F(t)) \wedge (M \geq \bullet F(t')) \Rightarrow (M \geq \bullet F(t) + \bullet F(t')),$$

т.е. если переход  $t$  может сработать, то никакой другой переход не может, сработав, лишиться этой возможности. Сеть  $N$  устойчива, если все ее переходы устойчивы.

В сети на рис. 8 устойчивы переходы  $t_1, t_2, t_3, t_4$ , в сети на рис. 9 устойчив только переход  $t_2$ . Однако обе сети не устойчивы, так как в первой сети не устойчивы переходы  $t_5, t_6$ , во второй –  $t_1, t_3, t_4$ .

Конечная цель специальной теории сетей Петри – автоматический анализ свойств сетей, их автоматический синтез и преобразования, на основании чего могут быть построены практические алгоритмы анализа, синтеза и преобразований дискретных систем, моделируемых сетями. В частности, полезно найти алгоритмы, с помощью которых для любой предъявленной сети можно установить, обладает ли она интересующим нас свойством – является ли она ограниченной, живой, устойчивой.

вой и т.п. В первую очередь нужно выяснить существование таких алгоритмов. Эти вопросы формулируются как массовые алгоритмические проблемы для сетей: проблема ограниченности (существует ли алгоритм, который позволяет узнать, является ли данная сеть ограниченной), проблема потенциальной живости переходов, проблема живости сетей, проблема устойчивости, проблема безопасности. Говорят, что проблема разрешима, если соответствующий алгоритм распознавания свойств существует, в противном случае проблема неразрешима.

Большинство из перечисленных выше проблем связано с определением возможности достижения некоторых специальных разметок в сети (например, с достижением  $t$ -тупиковых разметок для данного перехода  $t$ ), т.е. с проблемой достижимости заданной разметки. В этой проблеме ставится вопрос о существовании алгоритма, с помощью которого можно узнать для произвольной сети  $N$  и произвольной разметки  $M$ , принадлежит ли  $M$  множеству  $R(N)$ . Ниже будет показано, например, что проблемы живости и достижимости эквивалентны в этом смысле, что решение одной из них автоматически решает другую.

Несколько особняком стоят проблемы  $R$ -включения и  $R$ -эквивалентности сетей Петри. Пусть задан класс сетей, которые имеют одно и то же множество мест (или их множества мест изоморфны). В первом случае необходимо выяснить существование алгоритма, устанавливающего для любых двух сетей из этого класса, имеет ли место соотношение  $R(N_1) \subseteq R(N_2)$ , во втором –  $R(N_1) = R(N_2)$ .

### 3.2.2. Проблемы ограниченности и безопасности

Некоторое место  $p$  в сети  $N$  может оказаться неограниченным по двум причинам. Первая заключается в следующем. Пусть свободный язык сети  $L(N)$  содержит последовательность срабатываний  $\tau = \tau_1 \tau_2$  такую, что

$$(M_0[\tau_1]M_1) \wedge (M_1[\tau_2]M_2) \wedge (M_2 \geq M_1) \wedge (M_2(p) > M_1(p)).$$

Поскольку  $M_2 \geq M_1$ , то любая подпоследовательность срабатываний, начинающаяся при разметке  $M_1$ , может повториться и начиная с разметки  $M_2$ . Это означает, что любая последовательность вида  $\tau_1 \tau_2^n$  также принадлежит  $L(N)$ , каким бы большим не было  $n$ . Следовательно, разметка места  $p$  может безгранично расти. Например, для места  $p_2$  в сети на рис. 7 существует  $\tau = \lambda t_1$ , где  $\lambda$  – пустое слово, такое, что  $(M_0 = (1, 0, 0, 0)[t_1]M_1 = (1, 1, 0, 0)) \wedge (M_1 \geq M_0) \wedge (M_1(p_2) = 1 > M_0(p_2) = 0)$ .

Однако место  $p_4$  не ограничено по другой причине, так как нельзя найти пару достижимых в этой сети разметок  $M_1$  и  $M_2$  таких, что  $(M_2 \geq M_1) \wedge (M_2(p_4) > M_1(p_4))$ , хотя и ясно, что последовательности срабатываний вида  $t_1^n t_2 t_3^n$  ведут к накоплению любого числа  $n$  фишек в  $p_4$ . Можно отметить следующую разницу в неограниченности мест  $p_2$  и  $p_4$ : в  $p_2$  может быть "бесконечное" число фишек, а в  $p_4$  – сколько угодно большое, но конечное число фишек, так как переход  $t_3$  может сработать только конечное число раз, не большее, чем число срабатываний перехода  $t_1$ . Таким образом, неограниченность места  $p_4$  "вторична" по отношению к неограниченности места  $p_2$ .

Исследование проблемы ограниченности сводится к анализу графа – *покрывающего дерева* сети. Для любой сети такой граф конечен и может быть построен с помощью следующей процедуры:

- Первоначально предполагается, что дерево содержит единственную вершину-корень  $M_0$  и не имеет дуг.
- Путь  $M$  – вершина дерева, которая еще не объявлена листом (т.е. вершиной, из которой не исходит ни одна дуга), но в дереве нет исходящих из нее дуг. Возможны четыре случая:
  - 1) ни один из переходов сети не может сработать при разметке  $M$ , т.е.  $\forall t \in T : M \not\geq \bullet F(t)$ . В этом случае вершина  $M$  объявляется листом;
  - 2) на пути из корня дерева в вершину  $M$  существует вершина  $M'$  такая, что  $M' = M$ . Вершина  $M$  объявляется листом;
  - 3) на пути из корня дерева в вершину  $M$  существует вершина  $M'$  такая, что  $M' < M$ . Для любого места  $p$  такого, что  $M'(p) < M(p)$ , значение, соответствующей координаты  $M$  заменяется на  $\omega$  и вершина  $M$  объявляется  $\omega$ -листом;
  - 4) если ни один из вышеперечисленных случаев не имеет места, то  $M$  – внутренняя вершина дерева. Для каждого перехода  $t \in T$  такого, что  $M \geq \bullet F(t)$ , в дерево добавляется новая вершина  $M' = M - \bullet F(t) + F \bullet(t)$  и дуга, ведущая из  $M$  в  $M'$ , помеченная символом  $t$ .

#### Вопросы для самоконтроля

1. Для решения какого класса задач предназначены сети Петри?
2. Дайте формальное определение сети Петри.
3. Что такое разметка сети Петри?
4. Что такое место в сети Петри?
5. Что такое переход в сети Петри?
6. В каком случае переход называется потенциально живым?
7. Какая сеть Петри считается безопасной?

#### 4. ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ВЫПОЛНЕНИЯ БИЗНЕС-ПРОЦЕССОВ ПО ИХ ЖУРНАЛАМ ВЫПОЛНЕНИЯ

В главе описываются алгоритмы, помогающие аналитику проследить актуальное выполнение бизнес-процессов в системе электронного документооборота (СЭД), отражая аспекты реального поведения бизнес-процесса при выполнении (в результате выполнения бизнес-процесса проходит через множество точек выбора), выявляя незафиксированные формально правила принятия решений при прохождении через множество точек выбора, помогая отследить изменение этих правил со временем. Представленные алгоритмы базируются на технике интеллектуального анализа, расширив их для конкретной предметной области и улучшив показатели производительности.

Полученные в результате работы алгоритмов знания могут быть использованы для решения проблем:

- Предсказание успешного завершения экземпляра бизнес-процесса. Учитывая нынешнее состояние выполнения задач в экземпляре бизнес-процесса и используя статистику по выполнению модели автоматизированного бизнес-процесса (МАБП) в прошлом можно, отфильтровав последовательности по списку текущих задач и применив алгоритм поиска частых подпоследовательностей, ответить на этот вопрос.
- Идентификация критических задач. Во многих МАБП некоторые задачи могут быть рассмотрены как *критические*, в смысле планирования их выполнения системы управления потоками работ (Workflow systems, WFMS) в каждом удачном случае выполнения. Как правило, администратор знает на основе личного опыта какая из задач является критической в МАБП, однако не исключены случаи необходимости использования статистики актуального выполнения системы для прояснения этого вопроса.
- Характеристика неудачного/успешного выполнения. Анализируя предыдущее выполнение, аналитик может поинтересоваться какие дискриминантные факторы характеризуют удачное или неудачное выполнение бизнес-процесса.
- Улучшение МАБП. Информация, собранная в журнале выполнения МАБП, может быть выгодно использована для "оптимизации" выполнения. Например, критерий оптимальности может быть зафиксирован на каком-нибудь интересном параметре, таком как качество сервиса или среднее время завершения МАБП.

##### 4.1. РАЗРАБОТКА АЛГОРИТМА ПОЛУЧЕНИЯ КОМПЛЕКСНЫХ МАБП

Модели, основанные на графах, широко используются в большом спектре прикладных задач как интуитивно понятный и в то же время формальный способ представления разного рода данных, таких как web-документы, химическое строение, модели процессов, поведенческие шаблоны. Применение теории графов для анализа МАБП общепризнано и т.п.

Пример такой модели бизнес-процесса, описывающей обработку заявок от клиентов, представлен на рис. 10. (Пример построен на основе реального бизнес-процесса компании "Softintegro"). Опишем задачи, участвующие в модели процесса:

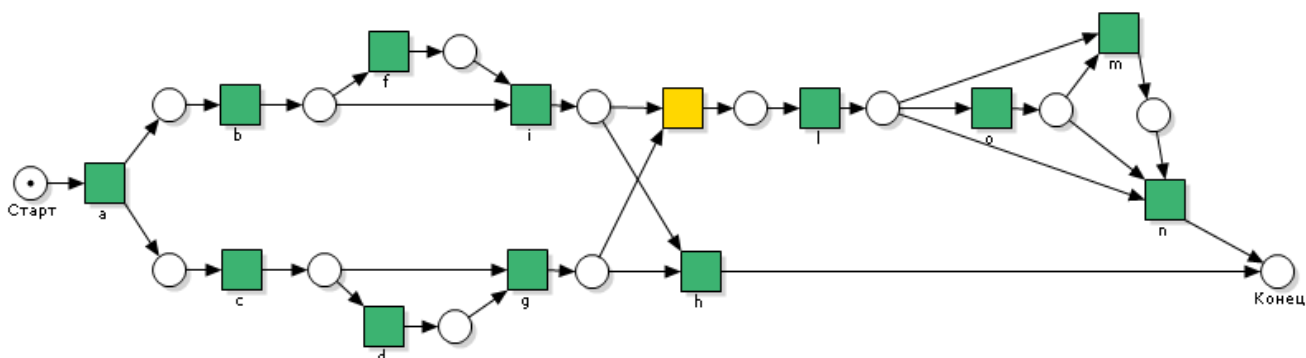


Рис. 10. МАБП "Обработка заказа от клиента"

- a – "Регистрация заказа";
- b – "Аутентификация клиента";
- c – "Проверка наличия";
- d – "Запрос к поставщику";
- f – "Регистрация нового клиента";
- g – "План производства";
- h – "Отклонение заказа";
- i – "Надежность клиента";
- l – "Принятие заказа";
- m – "Назначение скидки".
- n – "Подготовка счета";
- o – "Ускорение";

При выполнении такого процесса обнаруживается множество ограничений, например, в каждом экземпляре процесса, в котором участвовала задача b должна появиться задача i, или если выполнена задача l, то предшествовать ее выполнению должны задачи i и g, что очевидно с точки зрения бизнес-логики: если в бизнес-процессе происходила аутентификация клиента, то должен быть выполнен пункт "надежность клиента"; если в бизнес-процессе была выполнена задача "принятие заказа", то до этого обязательно были выполнены задачи "план производства" и "надежность клиента".

Таким образом, WF-модель описывает выполнение первоначального бизнес-процесса, выполнение которого может быть запротоколировано в БД.

## 4.2. ОПРЕДЕЛЕНИЕ ТЕРМИНОВ

Определим понятие журнала выполнения бизнес-процесса. Пусть  $T = \{A, B, C, \dots\}$  – множество задач бизнес-процесса, тогда нумерованный список элементов этого множества мы будем называть *последовательностью* (или *следом*), обозначаемой как  $\alpha(\alpha_1 \rightarrow \alpha_2 \rightarrow \dots \rightarrow \alpha_n)$ . Некоторое множество последовательностей  $L$  мы будем называть журналом выполнения автоматизированного бизнес-процесса.

Помимо приведенного определения журнала выполнения автоматизированного бизнес-процесса в работе потребуется понятие подпоследовательности. Последовательность  $\alpha$  является *подпоследовательностью*  $\beta$  (обозначается как  $\alpha \prec \beta$ ) тогда и только тогда, когда существуют числа  $i_1 < i_2 < \dots < i_n$  такие, что  $\alpha_i = \beta_{i_j}$  для всех  $\alpha_j$ .

Для алгоритма восстановления комплексных МАБП понятие комплексной WF-модели является ключевым.

**Определение 1.** Пусть  $P$  – это процесс. *Комплексная WF-модель* для  $P$  определяется как  $WS(P)$  – множество  $\{WS^1, \dots, WS^m\}$  WF-моделей для  $P$ . Размер  $WS(P)$  определяется как  $|WS(P)|$  – количество WF-моделей, содержащихся в ней.

Заметим также, что если последовательность в журнале принадлежит некоторой модели из множества комплексной схемы, то она также принадлежит и комплексной WF-модели.

**Определение 2.** Пусть дан след  $s$  журнала  $L$ , тогда WF-модель  $I = \alpha + (s)$  (где  $\alpha +$  – альфа плюс алгоритм ван-дер-Аальста) будем называть *экземпляром*.

Во многих исследованиях, даже несмотря на различия используемого в них синтаксиса, подразумевается (хоть и не указывается явно), что локальные ограничения бизнес-процессов могут быть выражены с использованием трех функций  $IN$ ,  $OUT_{\min}$  и  $OUT_{\max}$ , сопоставляющих каждому узлу число:

- $\forall a \in A - \{a_0\}, 0 < IN(a) < InDegree(a);$
- $\forall a \in A - F, 0 < OUT_{\min}(a) \leq OUT_{\max}(a) \leq OutDegree(a);$
- $IN(a_0) = 0$  и  $\forall a \in F, OUT_{\min}(a) = OUT_{\max}(a) = 0,$

где  $InDegree(a) = |\{e = (b, a)\}|$ ,  $OutDegree(a) = |\{e = (a, b)\}|$  и  $e \in E$ .

Задача  $a$  может стартовать тогда, когда как минимум  $IN(a)$  ее предшественников были завершены. Две типичные ситуации: (1) если  $IN(a) = InDegree(a)$ , тогда  $a$  является *and-join* задачей, которая может быть выполнена только тогда, когда все ее предшественники были выполнены, и (2) если  $IN(a) = 1$ , тогда  $a$  является *or-join* задачей, которая может быть выполнена если хотя бы одна из ее предшественников была выполнена. После завершения задача  $a$  может запустить одно непустое множество исходящих дуг с мощностью между  $OUT_{\min}(a)$  и  $OUT_{\max}(a)$ . Если  $OUT_{\max}(a) = OutDegree(a)$ , тогда  $a$  называется *full-fork*, если  $OUT_{\min}(a) = OUT_{\max}(a)$ , тогда  $a$  называется *and-split*, которая активизирует все ее последующие задачи. Итак, если  $OUT_{\max}(a) = 1$ , тогда  $a$  называется *xor-split*.

Выше были описаны *локальные ограничения*, однако в моделях существуют и *глобальные ограничения*. Глобальные ограничения в отличие от локальных отображают отношения между задачами не обязательно связанными друг с другом посредством дуг. Такие ограничения богаче по своей природе и их представление зависит от частной прикладной задачи моделируемого бизнес-процесса. Часто они выражаются с использованием сложного формализма. Как пример глобального ограничения в процессе на рис. 10  $f \rightarrow \neg m$  показывает, что если задача  $f$  была выполнена, то задача  $m$  не может быть выполнена. В контексте предложенной примерной модели бизнес-процесса это означает, что если был зарегистрирован новый клиент, то к его заказу не может быть начислена скидка.

Пусть даны множества глобальных ( $C_G(P)$ ) и локальных ( $C_L(P)$ ) ограничений для процесса  $P$ . Допустим дан подграф исходной WF-модели для процесса  $P - I$ , а также дано ограничение  $c \in C_L(P) \cup C_G(P)$ , мы записываем  $I | = c$ , если  $I$  удовлетворяет  $c$ . Более того, если  $I | = c$  для всех  $c$  в  $C_L(P) \cup C_G(P)$  и содержит стартовую задачу  $a_0$  и конечную задачу из  $F$ , тогда  $I$  называется экземпляром процесса  $P$ .

Обычно журналы записываются посредством следов. Длина следа обозначается как  $length(s)$ , а множество всех задач в следе обозначим через  $tastks(s) = \bigcup_{1 \leq i \leq length(s)} s[i]$ . Тогда журнал для процесса  $P$ , обозначаемый как  $L_P$ , будет мультимножеством следов:  $L_P = [s | s \in A^*]$ .

Пусть дан след  $s$  в журнале  $L_P$ ,  $P$  – процесс и  $I$  экземпляр этого процесса. Тогда  $s$  соответствует процессу  $P$  посредством  $I$ , обозначается через  $P | =^I s$ , т.е. в  $s$  содержатся задачи из  $A_I$ , таким образом, что каждая  $(a, b) \in E_I$ ,  $i < j$  где  $s[i] = a$  и  $s[j] = b$ . Более того  $s$  является совместимой с  $P$ , обозначаемой как  $P | = s$ , если существует  $I$  с  $P | =^I s$ . Итак, ослабленная формулировка соответствия, которая не полагается на существовании экземпляра  $I$ , может быть определено как  $P \mapsto s$ .

### 4.3. АЛГОРИТМ ПОЛУЧЕНИЯ КОМПЛЕКСНЫХ МАБП

Конечной целью работы алгоритма является получение комплексной WF-модели, для некоторого данного изначально журнала выполнения  $L$ , которая могла бы генерировать этот журнал "полным" настолько, насколько это возможно. Эта интуитивная посылка используется для того, чтобы ввести два критерия, называемых *завершенность* и *непротиворечивость*.

**Определение 3.** Под *непротиворечивостью* будем понимать  $s(WS, L) = \frac{|\{I \mid I \models WS \wedge \neg \exists s \in L\}|}{|\{I \mid I \models WS\}|}$ , т.е. процент последовательностей, принадлежащих обобщенной WF-модели и не имеющих соответствующих последовательностей в журнале выполнения. Под *завершенностью* будем понимать  $q(WS, L) = \frac{|\{s \mid s \in L \wedge s \models WS\}|}{|\{s \mid s \in L\}|}$ , т.е. процент последовательностей имеющих в журнале выполнения бизнес-процесса.

Легко видеть, что комплексная WF-модель состоящая всего из одного процесса содержит каждую последовательность в журнале бизнес-процесса, а это не совсем то, чего мы хотели бы достичь. Поэтому введем теперь еще одно ограничение – ограничение на размер комплексной модели бизнес-процесса.

**Определение 4 (Математическая постановка задачи).** Пусть  $L$  – это журнал процесса  $P$ . Даны три натуральных числа  $q$ ,  $m$  и  $n$ , задача обнаружения комплексной WF-модели, обозначаемой как  $MDP(P, q, m)$ , состоит в поиске  $q$ -завершенной комплексной WF-модели, такой что  $n \leq |WS| \leq m$ , где значение  $s(WS, L)$  минимально.

Иными словами мы собираемся предоставить задачу поиска  $q$ -завершенной комплексной WF-модели  $WS$  ( $n \leq |WS| \leq m$ ), которая непротиворечива настолько, насколько это возможно. Ниже приводится алгоритм для решения этой проблемы:

**Вход:** натуральное число maxProps

**Выход:** WS-модель

**Метод:**

$CF_i(WS_0^1) := \text{minePrecedences}(L_p)$ ;

$WS^\vee := WS_0^1$ ;

Пока  $|WS^\vee| < m$  выполнить

$WS_i^j := \text{leastSound}(WS^\vee)$ ;

$WS^\vee := WS^\vee - \{WS_i^j\}$ ;

$F := \text{searchProps}(L(WS_i^j), q, \text{maxProps}, CF_q)$ ;

$R(WS_i^j) := \text{проекция}(L(WS_i^j), F)$ ;

$k := |F|$ ;

Если  $k > 1$  тогда

$j := \max\{j \mid WS_{i+1}^j \in WS^\vee\}$ ;

$\langle WS_{i+1}^{j+1}, \dots, WS_{i+1}^{j+k} \rangle := \text{k-means}(R(WS_i^j))$ ;

Для каждого  $WS_{i+1}^h$  выполнить

$WS^\vee := WS^\vee \cup \{WS_{i+1}^h\}$ ;

$CF := \text{aphaAlgoritm}(L(WS_{j+1}^h))$ ;

Иначе

$WS^\vee := WS^\vee \cup \{WS_i^j\}$ ;

Конец Если;

Конец Пока;

Вернуть  $WS^\vee$ ;

В конечном счете для получения комплексной схемы процесса  $P$  используем идею последовательного и инкрементального улучшения схемы путем извлечения некоторых глобальных зависимостей, применяемых для кластеризации последовательностей. Алгоритм начинает свое выполнение с построения графа – представления последовательностей. Алгоритмы извлечения таких графов широко представлены в литературе.

Каждая WF-модель  $WS_i^j$ , последовательно добавляется в комплексную WF-модель  $WS^\vee$ , где  $i$  – количество оставшихся шагов, а  $j$  – это идентификатор схемы, позволяющий производить различие схем среди схем одного уровня. Кроме того, обозначим через  $L(WS_i^j)$  множество последовательностей в кластере принадлежащих  $WS_i^j$ .

Алгоритм использует жадную эвристику: на каждом этапе предпочтение отдается схеме, которая наиболее выгодно может быть улучшена. На практике мы улучшаем в меньшей степени непротиворечивую схему среди уже обнаруженных.

Кластеризация осуществляется путем метода  $k$ -средних, подготовительной фазой для которого выступают процедуры "поискСвойств" (обнаружение свойств для кластеризации) и "проекция" (для представления последовательностей в удобном для метода  $k$ -средних формате).

Рассмотрим процедуру "поискСвойств" более подробно. В некоторых работах уже сталкивались с подобной задачей, однако наша цель представить алгоритм нахождения таких свойств, которые были бы применимы именно в контексте получения комплексных WF-моделей.

Нам важны свойства, которые разделяют журнал наиболее выразительным образом. Для решения этой задачи введем еще ряд определений.

Последовательность задач  $[\alpha_1 \rightarrow \dots \rightarrow \alpha_h]$  является  $l$ -частой в журнале  $L$  если  $\frac{|\{\alpha \in L \mid \alpha_{i_1}, \dots, \alpha_{i_h} \wedge i_1 < \dots < i_h\}|}{|L|} \geq l$ . Мы говорим, что  $[\alpha_1 \rightarrow \dots \rightarrow \alpha_h]$   $l$ -предшествует  $\alpha$  в  $L$ , обозначаемой как  $[\alpha_1 \rightarrow \dots \rightarrow \alpha_h] \rightarrow_l \alpha$ , если обе последовательности  $[\alpha_1 \rightarrow \dots \rightarrow \alpha_h]$  и  $[\alpha_1 \rightarrow \dots \rightarrow \alpha_h \rightarrow \alpha]$  являются  $l$ -частыми.

**Определение 5 (Свойства).** Свойство  $\beta$  – это выражение вида  $[\alpha_1 \rightarrow \dots \rightarrow \alpha_h] \mapsto \alpha$  такое, что (1)  $[\alpha_1 \rightarrow \dots \rightarrow \alpha_h]$   $l$ -частое свойство в  $L$ , (2)  $[\alpha_1 \rightarrow \dots \rightarrow \alpha_h] \rightarrow_l \alpha$  не выполняется.

На основе этих определений представим алгоритм поиска наиболее дискриминирующих свойств:

**Вход:** Журнал выполнения  $L$ , величина  $q$ , максимальное количество свойств  $maxProps$ , граф зависимостей  $CF_q$

**Выход:** Множество минимальных свойств

**Метод:**

$$L_2 := \{[\alpha_1 \rightarrow \alpha_2] \mid (\alpha_1, \alpha_2) \in E_q\};$$

$$k := 1, R := L_2; F := \emptyset;$$

Повтор

$$M := \emptyset; k := k + 1;$$

Для всех  $[\alpha_1 \rightarrow \dots \rightarrow \alpha_j] \in L_k$  выполнить

Для всех  $[\alpha_j \rightarrow b] \in L_2$  выполнить

Если  $[\alpha_i \rightarrow \dots \rightarrow \alpha_j] \not\rightarrow b$  не в  $F$ , тогда

$$M := M \cup \{[\alpha_i \rightarrow \dots \rightarrow \alpha_j \rightarrow b]\};$$

Для всех  $p \in M$  выполнить

Если  $p$  – это  $l$ -частый в  $L$ , тогда

$$L_{k+1} := \{p\}$$

Иначе

$$F := F \cup \{[\alpha_i \rightarrow \dots \rightarrow \alpha_j] \not\rightarrow b\};$$

Конец Для всех;

$$R := R \cup L_{k+1};$$

Пока  $L_{k+1} = \emptyset$ ;

$$S' := L; F' := \emptyset;$$

Выполнить

$$\beta := \operatorname{argmax}_{\beta \in F} |w(\beta, S')|$$

$$F' := F' \cup \{\beta\};$$

$$S' := S' - w(\beta, S');$$

Пока  $(|S'| / |L_p| > q) \&\& (|F'| < maxProps)$ ;

Вернуть  $F'$ ;

Сначала на каждом этапе  $k$  выполнения мы добавляем в  $L_k$  все  $l$ -частые последовательности, чей размер равен  $k$ . И этот процесс повторяется до тех пор, пока не будут найдены все  $l$ -частые последовательности разных размеров. Далее это множество последовательностей фильтруется в соответствии с вводимыми при вызове метода ограничениями.

Представленный метод был реализован в программном прототипе и апробирован на реальных журналах выполнения бизнес-процессов.

#### 4.4. РАЗРАБОТКА АЛГОРИТМА ПОЛУЧЕНИЯ ЧАСТЫХ ПОДПОСЛЕДОВАТЕЛЬНОСТЕЙ

Рассмотрим проблему поиска наиболее повторяющихся шаблонов (т.е. подграфов в нашей терминологии) в экземплярах бизнес-процессов, заключающейся в нахождении наиболее повторяющихся фрагментов в бизнес-процессе при его выполнении.

Одной из разновидностей задачи поиска частых подпоследовательностей является задача ответа на вопрос: какие задачи из модели бизнес-процесса, представленного на рис. 10, выполняются в наиболее часто при выполнении задачи h? В качестве примера рассмотрим экземпляры:

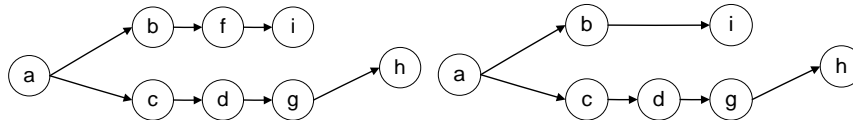


Рис. 11. Экземпляры для бизнес-процесса "Обработка заказа от клиента"

Отметим, что оба графа с рис. 11 являются подграфами первоначальной модели бизнес-процесса, показанного на рис. 10, и удовлетворяют ограничениям там указанным. Кроме того, оба подграфа соответствуют требованию на содержание узла h.

Рассмотрим подграф с рис. 12.

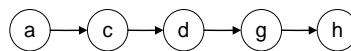


Рис. 12. Подграф для экземпляров с рис. 10

Часто выпадает при выполнении обоих экземпляров, и характеризуется выполнением задач c, d, g. В таком случае это может означать, что отказ от обслуживания часто характеризуется недостатком материалов на предприятии. С точки зрения анализа бизнеса становится очевидным необходимость внесения изменений в стратегию управления хранением комплектующих. Сложность решения задачи поиска наиболее частых шаблонов заключается в выборе алгоритма генерации желаемых паттернов путем "умного" изучения области поиска, на которые накладываются ограничения схемы.

При поиске частых подпоследовательностей WF-модель выступает в качестве стартовой точки исследования, а не результата: некоторое количество исполнений бизнес-процесса анализируется контекстуально и базируется на этой WF-модели с целью найти частые шаблоны выполняемых задач, таким образом исследуя полезную информацию для поддержки принятия решений в организации. С учетом этой перспективы исследование следует сопоставить с другими попытками анализа частых подпоследовательностей. Наиболее известны из этого класса исследований базирующиеся на свойстве *антимонотонности*. В этой работе представляется алгоритм "Apriori": общая идея заключается в генерации множества кандидатов  $k + 1$ , комбинируя в подходящем случае размер  $k$  множества частых шаблонов, и проверяя их частоту.

Совершенно иное исследование проблемы было предложено в [6], которое использует метод "FP-growth". По существу идея анализа частых шаблонов совпадает с вышеописанным, посредством рекурсивного проецирования базы данных на частые шаблоны уже найденные и затем комбинирование результатов исследования с проецированной базой данных. Расширение этой идеи для последовательных шаблонов предложено в [7]. Недавняя попытка комбинирования такого метода с алгоритмом Apriori была предпринята в [8].

Как и проблема интеллектуального анализа шаблонов в сложных предметных областях, исследование частых деревьев была произведена в [9] (алгоритм "AGM"), в то время, как первый алгоритм в этом направлении, базирующийся на свойстве антимонотонности, был предложен в [10]. Отметим, что эта задача все еще обещает интересные задачи в различных предметных областях, таких как биоинформатика и анализ web-документов.

Например, поиск используемый в "AGM" был адаптирован и улучшен в алгоритме "FSG" [11], в котором используется умная стратегия для маркировки сгенерированных графов с целью уменьшить вычислительные расходы на проверку изоморфности графов. Более того, в настоящее время появляются новые алгоритмы, основанные на методе проекции: "gSpan" [12] исследует все подграфы без генерации кандидатов и использует позитивное усечение, в то время как "CloseGraph" [13] сокращает количество шаблонов, подлежащих генерации за счет использования понятия *закрытых* шаблонов, т.е. шаблонов, которые не являются подграфами для других шаблонов с той же поддержкой.

Понятно, что подобные подходы могут быть использованы для решения проблемы поиска наиболее частых последовательностей для WF-моделей, после адаптации этих алгоритмов с учетом специфики данного прикладной области применения.

Тем ни менее адаптация вышеописанных методов к поиску частых подпоследовательностей в автоматизированных бизнес-процессах – сложная задача. Действительно, генерация шаблонов посредством таких традиционных подходов не учитывает тех ограничений, которые накладывает сама схема, таких как взаимоотношения между задачами, синхронизация и параллельное выполнение задач (см. [14] – [16]). В противоположность этому алгоритмы, представленные ниже, учитывают эти ограничения, накладываемые структурой WF-модели. А экспериментальные результаты представленные далее показывают, что они превосходят традиционные подходы к восстановлению частых подпоследовательностей.



#### 4.4.1. Алгоритм поиска частых подпоследовательностей "f-поиск"

Определим необходимые в дальнейшем понятия. Допустим у нас есть WF-модель  $P$  и множество экземпляров  $F = \{I_1, \dots, I_n\}$ . Граф  $p = \langle A_p, E_p \rangle \subseteq P$  называется  $F$ -шаблоном, если существует  $I = \langle A_I, E_I \rangle \in F$  такое, что  $A_p \subseteq A_I$  и  $p$  подграф  $I$ , включенный посредством узлов в  $A_p$ .

Допустим  $freq(p) = \{I \in F \mid I \supseteq p\} / |F|$  поддержка  $F$ -шаблона  $p$ . Тогда мы можем дать следующее определение:

**Определение 6 (Математическая формулировка задачи).**  $FPCPD(\sigma)$ : задача поиска всех связанных шаблонов, поддержка которых больше чем  $\sigma$ .

Если подходить к решению этой задачи прямолинейно, то напрашивается алгоритм поиска шаблонов на основе простой генерации прямо связанных подграфов, а затем тестирование в полиномиальном времени является ли этот шаблон экземпляром для  $P$ . Другое предложение основано на идее ослабления количества шаблонов, подлежащих генерации. Для того чтобы добиться этой цели, мы можем рассматривать только графы, которые "закрыты" (с учетом глобальных и локальных ограничений), т.е. таких, что  $p|_c = c$  для всех  $c \in C_L \cup C_G$ . Будем называть такие графы *ослабленными шаблонами* или просто  $w$ -шаблоны. Формализуем приведенные выше рассуждения:

**Определение 7.** Дана WF-модель  $p' = \langle A_{p'}, E_{p'} \rangle$ , *детерменестическое закрытие*  $p$  ( $ws\_закрытие(p)$ ) определяется, как граф  $p' = \langle A_{p'}, E_{p'} \rangle$  такой, что: (1)  $A_p \subseteq A_{p'}$ , и  $E_p \subseteq E_{p'}$ , (2)  $a \in A_{p'}$  – *and-join* подразумевает, что для каждого  $(b, a) \in E$ ,  $(b, a) \in E_{p'}$  и  $b \in A_{p'}$ , (3)  $a \in A_{p'}$  – *deterministic fork* подразумевает, что для каждого  $(a, b) \in E$  с  $b$  – *or-join*,  $(a, b) \in E_{p'}$  и  $b \in A_{p'}$ . Более того граф  $p$  такой, что  $p = ws\_закрытие(p)$  называется  $ws\_закрытым$ .

**Определение 8.** *Ослабленный шаблон*, или просто  $w$ -шаблон, это  $ws\_закрытый$  граф  $p$ , такой что для каждого узла  $a$ ,  $|\{(a, b) \mid (a, b) \in E_p\}| \leq OUT_{\max}(a)$ .

**Определение 9.** Пусть дана WF-модель  $WS = \langle A, E \rangle$ , тогда для каждого  $a \in A$  граф  $p = ws\_закрытие(\langle \{a\}, \{\} \rangle)$  называется элементарным  $w$ -паттерном.

Для поиска решения задачи используется алгоритм, который инкрементально строит частые  $w$ -шаблоны, стартуя с "элементарных"  $w$ -шаблонов (описанных ниже) и расширяя каждый частый шаблон  $max$  посредством использования двух базовых операций: добавление частой дуги и слияние с другим частым элементарным  $w$ -шаблоном.

Элементарные  $w$ -шаблоны, с которых начинается построение частых шаблонов, получаются как детерменестическое закрытие отдельных узлов. Шаблон является элементарным  $w$ -шаблоном (обозначим через  $ew$ -шаблон) для узла  $a$ , если это минимальный  $w$ -шаблон содержащий  $a$ . Множество всех  $ew$ -шаблонов обозначается как  $EW$ . Более того, пусть  $p$   $w$ -шаблон, тогда  $EW_p$  обозначает множество элементарных  $w$ -шаблонов содержащихся в  $p$ . Заметим, что если дан  $ew$ -шаблон  $e$ ,  $EW_e$  не обязательно содержит только один элемент, может содержать другие  $ew$ -шаблоны. Кроме того, данное множество  $E' \subseteq EW$ ,  $Compl(E') = EW - \bigcup_{e \in E'} EW_e$  содержит все элементарные шаблоны, которые не содержатся ни в  $E'$  ни в одном элементе  $E'$ .

Теперь введем операцию отношения  $<$  между связанными  $w$ -шаблонами. Определим через  $E^{\subseteq}$  множество всех дуг в  $P$ , источники которых не являются *and-split*, т.е.  $E^{\subseteq} = \{(a, b) \in E \mid OUT_{\min}(a) < OutDegree(a)\}$ . Даны два соединенных  $w$ -шаблона  $p = \langle A_p, E_p \rangle$  и  $p' = \langle A_{p'}, E_{p'} \rangle$ , мы говорим что  $p < p'$  тогда и только тогда, когда:

1.  $A_p = A_{p'}$  и  $E_{p'} = E_p \cup \{(a, b)\}$ , где  $(a, b) \in E^{\subseteq} - E_p$  и  $OUT_{\max}(a) > OutDegree_p(a)$  (т.е.  $p'$  может быть получена из  $p$  посредством добавления дуги).
2. Существует  $p'' \in Compl(EW_p)$  такая, что  $p' = p \cup p'' \cup X$ , где  $X$  либо пустое множество, если  $p$  и  $p''$  связанные или содержит дугу в  $E^{\subseteq}$  с конечными точками в  $p$  и  $p''$  (т.е.  $p'$  получено из  $p$  путем добавления элементарного  $w$ -шаблона и возможно соединяющей дуги.)

При работе алгоритма инкрементально строятся частые  $w$ -шаблоны с использованием двух базовых операций: добавление частой дуги и слияние с другим частым  $w$ -шаблоном.

Элементарные шаблоны с которых начинается алгоритм получаются как  $ws\_закрытие(p)$  для каждого из узлов. Далее идет вычисления в главном цикле алгоритма, где каждое новое значение для  $L_{k+1}$  получается путем расширения любого шаблона  $p$  сгенерированного на предыдущем шаге ( $p \in L_k$ ) двумя способами: 1) добавлением частой дуги из  $E^{\subseteq}$  (посредством функции  $addFrequentArc$ ) и 2) добавлением элементарного  $w$ -шаблона (функция  $AddFrequentEWPatten$ ). Каждый шаблон  $p'$ , генерируемый функциями, указанными выше, – допустимый подграф для  $WS$ , т.е. для каждой  $a \in A_{p'}$ ,  $OutDegree_{p'}(a) \leq OUT_{\max}(a)$ .

Приведем псевдокод алгоритма:

**Вход:**  $WF$ -модель  $WS$ , множество экземпляров  $F = \{I_1, \dots, I_N\}$  для модели  $WS$

**Выход:** Множество частых  $F$ -шаблонов

**Метод:**

$L_0 := \{e \mid e \in EW, e \text{ является частым шаблоном с учетом } F\};$

$k := 0; R := L_0;$

$FrequentArcs := \{(a, b), (a, b) \in E^{\subseteq}, \langle a, b \rangle, \{(a, b)\} \text{ является частым с учетом } F \rangle\}$

$E_{\bar{f}}^{\subseteq} := E^{\subseteq} \cap FrequentArcs;$

Повтор

$U := 0;$

Для всех  $p \in L_k$  выполнить

$U := U \cup addFrequentArc(p);$

Для всех  $e \in Compl(EW_p) \cap L_0$  выполнить

$U := U \cup addFrequentEWPatten(p, e);$

Конец Для всех;

$L_{k+1} := \{p \mid p \in U, p\text{-частый с учетом } F\};$

$R := R \cup L_{k+1};$

Пока  $L_{k+1} = 0;$

Вернуть  $R;$

Процедура  $addFrequentEWPatten(p = \langle A_p, E_p \rangle, e = \langle A_e, E_e \rangle): w$ -шаблон;

$p' := \langle A_p \cup A_e, E_p \cup E_e \rangle;$

Если  $p'$  связанный, тогда Вернуть  $p'$

Иначе Вернуть  $addFrequentConnection(p', p, e);$

Процедура  $addFrequentConnection(p' = \langle A_{p'}, E_{p'} \rangle, p = \langle A_p, E_p \rangle, e = \langle A_e, E_e \rangle$

$S := 0;$

Для всех  $frequent(a, b) \in E_{\bar{f}}^{\subseteq} - E_p, (a \in A_p, b \in A_e) \vee (a \in A_e, b \in A_p)$  выполнить

$q := \langle A_{p'}, E_{p'} \cup (a, b) \rangle;$

Если  $WS \models q$ , тогда  $S := S \cup \{q\};$

Конец Для всех;

Вернуть  $S;$

Процедура  $addFrequentArc(p = \langle A_p, E_p \rangle):$  шаблон

$S := 0;$

Для всех  $frequent(a, b) \in E_{\bar{f}}^{\subseteq} - E_p, a \in A_p, b \in A_p$  выполнить

$p' := \langle A_p, E_p \cup (a, b) \rangle;$

Если  $WS \models p'$ , тогда  $S := S \cup \{p'\};$

Конец Для всех

Вернуть  $S;$

#### 4.4.2. Алгоритм поиска частых подпоследовательностей ("s-поиск")

Заметим, что при поиске частых подпоследовательностей можно использовать другую стратегию: итеративно генерировать кандидата на  $n$ -м уровне путем слияния кандидатов на  $j$ -м и  $n-j$  уровне, соответственно. Ясно, что в худшем случае (для  $j = n - 1$ ) мы получим алгоритм поиска, представленный в предыдущем разделе, иначе, в самом лучшем случае поиск сходится экспоненциально меньшим количеством итераций. Также ясно, что нам необходимо дополнительные усилия для идентификации компонент для слияния. Грубо говоря, эти компоненты должны быть таковы, что их границы могут быть сопоставлены. Под границей понимается множество узлов, которые допускают входящую или исходящую дугу.

Формализуем приведенные выше рассуждения.

**Определение 10.**  $BX\_ГРАНИЦА(p) = \{a \in A_p \mid InDegree_p(a) < InDegree(a)\}$  – множество узлов в шаблоне  $p$ , которые допускают дальнейшие входящие дуги,  $ISX\_ГРАНИЦА(p) = \{a \in A_p \mid OutDegree_p(a) < OUT_{\max}(a)\}$  – множество уз-

лов в шаблоне  $p$ , которые допускают наличие исходящих дуг. Множества  $VX\_ГРАНИЦА(p)$  и  $ИСХ\_ГРАНИЦА(p)$  представляют входящие и исходящие границы шаблона  $p$ , т.е. множество узлов внутри  $p$ , которые могут быть "достигнуты" другими узлами извне. Заметим, что по определению входящая граница для  $w$ -шаблона не может содержать *and-join* задачи. Аналогично, исходная граница не может содержать *deterministic fork*.

Границы могут быть использованы для соединения компонент. Так как дуга соединяет границы двух компонент, достаточно уделить все внимание частым дугам и итеративно генерировать новые кандидаты посредством слияния частых компонент, чьи границы соединены посредством этих дуг.

На основе этих идей был разработан алгоритм ("s-поиска"), подробности которого приводятся ниже:

**Вход:** WF-модель  $WS$ , множество экземпляров  $F = \{I_1, \dots, I_N\}$  для модели  $WS$

**Выход:** Множество частых  $F$ -шаблонов

**Метод:**

$R := \{e \mid e \in EW, e \text{ является частым шаблоном с учетом } F\}; \Delta R := R;$

Для всех  $(a, b) \in E$  выполнить  $connected\_by(a, b) = \emptyset;$

Повторить

Для всех  $a \in A$  выполнить

$INF(a) := \{p \in R \mid a \in VX\_ГРАНИЦА(p)\}, INF_P(a) = \emptyset;$

$OUTF(a) := \{p \in R \mid a \in ISX\_ГРАНИЦА(p)\}, OUTF_P(a) = \emptyset;$

Конец Для всех;

$FA := \{(a, b) \mid (a, b) \text{ является частой дугой с учетом } F, OUTF(a) \neq \emptyset, INF(b) \neq \emptyset\};$

$FP := \{p \cup q \mid p \cap q \neq \emptyset, p \in R, q \in \Delta R, WS \models p \cup q\};$

Для всех  $(a, b) \in FA$  выполнить

Для всех  $p_1 \in OUTF(a), p_2 \in INF(b)$  с учетом  $(a, b) \notin p_1 \cup p_2$  и  $(p_1, p_2) \notin connected\_by(a, b)$  выполнить

$q := p_1 \cup p_2 \cup \{(a, b)\};$

Если  $WS \models q$ , тогда

$FP := FP \cup \{q\};$

$VX\_ГРАНИЦА(q) := ComputeInBorder(b, p_1, p_2);$

$ИСХ\_ГРАНИЦА(q) := ComputeOutBorder(a, p_1, p_2);$

Для всех  $a \in VX\_ГРАНИЦА(q)$  выполнить  $INF_P(a) := INF(a) \cup \{q\};$

Для всех  $a \in ISX\_ГРАНИЦА(q)$  выполнить  $OUTF_P(a) := OUTF_P(a) \cup \{q\};$

$connected\_by(a, b) := connected\_by(a, b) \cup \{(p_1, p_2)\}$

Конец Если;

Конец Для всех;

$\Delta R := \{p \in FP \mid p\text{-частый с учетом } F\};$

$R := R \cup \Delta R;$

Пока  $\Delta R = \emptyset;$

Вернуть  $R;$

Процедура  $ComputeInBorder(b, p_1, p_2);$

Если  $|InDegree_{p_1 \cup p_2}(b)| < InDegree(b) - 1$ , тогда

$VX\_ГРАНИЦА := \{b\};$

Иначе  $ГРАНИЦА := \emptyset;$

Для всех  $c \in (VX\_ГРАНИЦА(p_1) \cup VX\_ГРАНИЦА(p_2)) - \{b\}$  выполнить

Если  $|InDegree_{p_1 \cup p_2}(c)| < InDegree(b)$ , тогда  $VX\_ГРАНИЦА := VX\_ГРАНИЦА \cup \{c\};$

Вернуть  $VX\_ГРАНИЦА;$

Процедура  $ComputeOutBorder(a, p_1, p_2);$

Если  $|OutDegree_{p_1 \cup p_2}(a)| < OUTF_{max}(a) - 1$ , тогда  $ИСХ\_ГРАНИЦА := \{a\}$ , тогда  $ИСХ\_ГРАНИЦА := \emptyset;$

Для всех  $c \in (ИСХ\_ГРАНИЦА(p_1) \cup ISX\_ГРАНИЦА(p_2)) - \{a\}$  выполнить

Если  $|OutDegree_{p_1 \cup p_2}(c)| < OUTF_{max}(a)$ , тогда  $ИСХ\_ГРАНИЦА := ISX\_ГРАНИЦА \cup \{c\};$

Вернуть  $ИСХ\_ГРАНИЦА;$

Ядром алгоритма является главный цикл, в котором для каждого узла  $a \in WS$  множества  $INF(a)$  (или  $OUTF(a)$ )  $F$ -шаблонов, содержащих  $a$  на входе (или выходе) вычисляются границы. Далее переменные  $FA$  и  $FP$  заполняются значениями частых дуг, которые могут соединить паттерны и кандидатами, которые могут быть получены составлением "совмес-

тимых" шаблонов. Далее границы пересчитываются для новых  $F$ -паттернов, и частые  $F$ -паттерны идентифицируются вычислением частоты каждого кандидата. Заметим, что границы для кандидата  $F$ -паттерна может быть инкрементально вычислена посредством расширения границ связанных компонент, и новые кандидаты могут быть генерированы путем слияния  $F$ -паттернов, открывая доступ к некоторым узлам. Алгоритм завершает работу, когда количество кандидатов равно нулю, т.е. когда вычисленные паттерны имеют пустые входные и выходные границы.

#### **Вопросы для самоконтроля**

1. Дайте определение комплексной МАБП.
2. Дайте определение журнала выполнения бизнес-процесса.
3. Что такое подпоследовательность?
4. Что такое  $w$ -шаблон?
5. В каком случае алгоритм  $s$ -поиска будет превосходить по производительности алгоритм  $f$ -поиска?

#### **ЗАКЛЮЧЕНИЕ**

Трудности, с которыми приходится сталкиваться при разработке сложных МАБП, стимулируют разработку технологий восстановления моделей автоматизированных бизнес-процессов и интеллектуального анализа их выполнения в целом, цель которого заключается в автоматическом получении модели процесса на основе аккумулированных данных. Существующие техники позволяют получить МАБП, хорошо подходящие для применения в WFMS.

Несмотря на большой рост интереса к методам и алгоритмам интеллектуального анализа выполнения МАБП все еще не существует системы, позволяющей их использовать при реальной работе аналитика. Именно поэтому в учебном пособии представлены алгоритмы интеллектуального анализа, позволяющие реализовать приведенные в обзоре методы и устранить сложности внедрения и использования СЭД.

## СПИСОК ЛИТЕРАТУРЫ

1. Ляпин, Н.Р. Восстановление моделей бизнес-процессов: проблемы и подходы из решения / Н.Р. Ляпин, Б.С. Дмитриевский // Глобальный научный потенциал : сб. материалов 3-й Междунар. науч.-практ. конф., г. Тамбов, 23-24 апр. 2007 г. / Тамб. гос. техн. ун-т (ТГТУ). – Тамбов, 2007. – С. 100 – 103.
2. Ляпин, Н.Р. Интеллектуальный анализ выполнения бизнес-процессов в системе электронного документооборота / Н.Р. Ляпин // Теория, методы проектирования, программно-техническая платформа корпоративных информационных систем : материалы V Междунар. науч.-практ. конф., г. Новочеркасск, 25 мая 2007 г. / Юж.-Рос. гос. техн. ун-т. – Новочеркасск : ЮРГТУ, 2007. – С. 53 – 55.
3. Ляпин, Н.Р. Разработка информационной системы восстановления моделей автоматизированных бизнес-процессов / Н.Р. Ляпин, Б.С. Дмитриевский // Программные продукты и системы. – 2007. – № 3(79). – С. 80 – 81.
4. Ляпин, Н.Р. Получение комплексных моделей бизнес-процессов на основе журналов их выполнения / Н.Р. Ляпин, Б.С. Дмитриевский // Телекоммуникационные и информационные системы : тр. междунар. конф. – СПб., 2007. – С. 259 – 265.
5. Ляпин, Н.Р. Автоматизация делопроизводства как инструмент повышения качества управления производством / Н.Р. Ляпин, Б.С. Дмитриевский // Компьютерные технологии в науке, производстве, социальных и экономических процессах : материалы IV Междунар. науч.-практ. конф., г. Новочеркасск, 14 нояб. 2003 г. / Юж.-Рос. гос. техн. ун-т. – Новочеркасск, 2003. – Ч. 4. – С. 33–34.
6. Pei, J. PrefixSpan: Mining Sequential Patterns Efficiently by PrefixProjected Pattern Growth / J. Pei, J. Han, B. Mortazavi-Asl // Proc. 2001 Int. Conf. Data Engineering (ICDE'01), Heidelberg, Germany, April 2001. – P. 215 – 224.
7. Han, J. Mining frequent patterns without candidate generation / J. Han, J. Pei, Y. Yi // Proc. Int. ACM Conf. On Management of Data (SIGMOD'00), 2000. – P. 1 – 12.
8. Pei, J. H-Mine: Hyper-structure mining of frequent patterns in large databases / J. Pei, J. Han, H. Lu, S. Nishio, S. Tang, D. Yang // Proc. IEEE Int. Conf. on Data Mining (ICDM'01), 2001. – P. 441 – 448.
9. Zaki, M. Efficiently Mining Frequent Trees in a Forest / M. Zaki // Proc. 8<sup>th</sup> Int. Conf. On Knowledge Discovery and Data Mining (SIGKD02), 2002. – P. 71 – 80.
10. Inokuchi, A. An Apriori-Based Algorithm for Mining Frequent Substructures from Graph Data / A. Inokuchi, T. Washi, H. Moroda // Proc. 4<sup>th</sup> European Conf. on Principles of Data Mining and Knowledge Discovery, 2000. – P. 13 – 23.
11. Kuramochi, M. Frequent subgraph discovery / M. Kuramochi, G. Karypis // Proc. IEEE Int. Conf. on Data Mining (ICDM'01), 2001. – P. 313 – 320.
12. Yan, X. gSpan: Graph-Based Substructure Pattern Mining / X. Yan, J. Han // Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002), IEEE Computer Society, 2002. – P. 721 – 724.
13. Yan, X. CloseGraph: Mining closed frequent graph patterns / X. Yan, J. Han // Proc. ACM Int. Conf. on Knowledge Discovery and Data Mining (KDD'03), 2003. – P. 286 – 295.
14. Koksal, P. Workflow history management / S.N. Arpinar, A. Dogac // SIGMOD Record, 1998. – Vol. 27, issue 1. – P. 67 – 75.
15. Van der Aalst, W.M.P. Workflow Management: Models, Methods, and Systems. MIT Press. – 2002.
16. Cook, J.E. Automating process discovery through event-data analysis / J.E. Cook, A.L. Wolf // Proc of the 17<sup>th</sup> Conference on Software Engineering, Seattle, Washington, April 1995.
17. Van der Aalst, W.M.P. An Alternative Way to Analyze Workflow Graphs / W.M.P. Van der Aalst, A. Hirsenschall, H.M.W. Verbeek // Proceedings of the 14<sup>th</sup> International Conference on Advanced Information Systems Engineering (CAiSE'02), Lecture Notes in Computer Science 2348, 2002. – P. 535 – 552.

## ОГЛАВЛЕНИЕ

<b>ВВЕДЕНИЕ</b> .....	3
<b>1. СИСТЕМЫ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА</b> .....	5
1.1. Сущность управления документооборотом фирмы .....	5
1.2. Бизнес-процессы управления документооборотом .....	6
1.3. Недостатки традиционного подхода в управлении документооборотом .....	14
1.4. Технология потоков работ как принципиально новый подход к управлению предприятием .....	15
1.5. Список стандартов в области систем управления документооборотом .....	17
1.6. Описание архитектуры системы управления электронного документооборота .....	19
Вопросы для самоконтроля .....	21
<b>2. СИСТЕМА ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА DOCSVISION</b> .....	21
2.1. Принципы организации платформы DocsVison .....	23
2.2. Навигатор. Общие сведения .....	23
2.3. Список папок .....	25
2.4. Область просмотра содержимого папки .....	26
2.5. Область предварительного просмотра .....	26
2.6. Область группировки .....	26
2.7. Работа с папками .....	27
2.8. Работа с карточками .....	35
2.9. Работа с представлениями .....	43
2.10. Возможности поиска .....	43
2.11. Права доступа к объектам .....	47
2.12. Архивирование объектов .....	48
Вопросы для самоконтроля .....	49
<b>3. СЕТИ ПЕТРИ</b> .....	49
3.1. Формализация .....	57
3.2. Свойства сетей Петри и их анализ .....	61
3.2.1. Основные свойства сетей Петри .....	61
3.2.2. Проблемы ограниченности и безопасности .....	64
Вопросы для самоконтроля .....	66
<b>4. ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ВЫПОЛНЕНИЯ БИЗНЕС-ПРОЦЕССОВ ПО ИХ ЖУРНАЛАМ ВЫПОЛНЕНИЯ</b> .....	66
4.1. Разработка алгоритма получения комплексных МАБП .....	67
4.2. <i>Определение терминов</i> .....	69
4.3. Алгоритм получения комплексных МАБП .....	71
4.4. Разработка алгоритма получения частых подпоследовательностей .....	74
4.4.1. Алгоритм поиска частых подпоследовательностей "f-поиск" .....	76
4.4.2. <i>Алгоритм поиска частых подпоследовательностей ("s-поиск")</i> .....	80
Вопросы для самоконтроля .....	82
<b>ЗАКЛЮЧЕНИЕ</b> .....	83
<b>СПИСОК ЛИТЕРАТУРЫ</b> .....	84