

# Применение сетей Петри для управления рабочими процессами

W.M.P. van der Aalst

Факультет математики и вычислительной техники, Технологический университет Эйндховена, почтовый ящик 513, NL-5600 MB, Эйндховен, Нидерланды,  
телефон: +31 40 2474295, электронная почта: [wsinwa@win.tue.nl](mailto:wsinwa@win.tue.nl)

## Аннотация

*Управление рабочими процессами обещает новое решение извечной проблемы: управление, мониторинг, оптимизацию и поддержку бизнес-процессов. Что нового в управлении рабочими процессами, так это четкое представление логики бизнес-процесса, которое обеспечивает компьютеризированную поддержку. В этой статье обсуждается использование сетей Петри в контексте управления рабочими процессами. Сети Петри являются признанным инструментом для моделирования и анализа процессов. С одной стороны, сети Петри можно использовать в качестве языка проектирования для спецификации сложных рабочих процессов. С другой стороны, теория сети Петри предусматривает мощные методы анализа, которые могут быть использованы для проверки правильности процедур рабочего процесса. В этом документе представлено управление рабочими процессами как область применения сетей Петри, представлены современные результаты с учетом проверки рабочих процессов и освещены некоторые инструменты для управления рабочими процессами на основе сети Петри.*

## 1 Введение

В прежние времена информационные системы разрабатывались для поддержки выполнения отдельных задач. Современные информационные системы должны поддерживать текущие бизнес-процессы. Уже недостаточно сосредоточиваться только на задачах. Информационная система также должна контролировать и поддерживать логистические аспекты бизнес-процесса. Другими словами, информационная система также должна управлять потоком работы в организации. Многие организации со сложными бизнес-процессами определили потребность в концепциях, методах и инструментах для поддержки управления рабочими процессами. Исходя из этой потребности, термин появилось управление рабочими процессами (ср. [HL91, Kou95]). До недавнего времени не существовало универсальных инструментов для поддержки управления рабочими процессами. В результате части бизнес-процесса были жестко запрограммированы в приложениях. Например, достаточно того, что приложение для поддержки задачи X запускает другое приложение для поддержки задачи

У. Это означает, что одно приложение знает о существовании другого приложения.

Это нежелательно, потому что каждый раз, когда изменяется базовый бизнес-процесс, приложения необходимо модифицировать. Более того, аналогичные конструкции должны быть реализованы в нескольких приложениях, и невозможно отслеживать и контролировать весь рабочий процесс. Поэтому несколько поставщиков программного обеспечения признали необходимость в системах управления рабочими процессами. Система управления рабочими процессами (WFMS) – это универсальный программный инструмент, который позволяет определять, выполнять, регистрировать и контролировать рабочие процессы (см. [Law97]). На данный момент многие поставщики предлагают систему управления рабочими процессами. Это показывает, что индустрия программного обеспечения признает потенциал инструментов управления рабочими процессами.

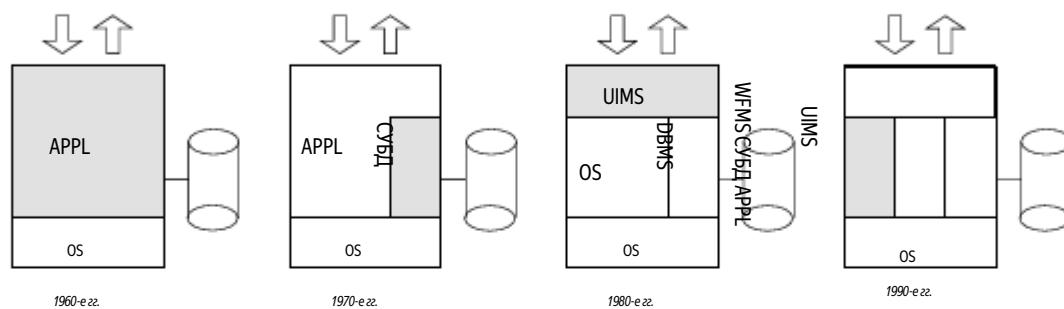


Рисунок 1: Системы управления рабочими процессами в исторической перспективе.

Чтобы лучше понять влияние управления рабочими процессами в ближайшем будущем, полезно рассмотреть эволюцию информационных систем за последние четыре десятилетия (см. [Aal96b]).

На рисунке 1 показан феномен управления рабочими процессами в исторической перспективе.

Рисунок иллюстрирует эволюцию информационных систем за последние четыре десятилетия, описывая архитектуру типичной информационной системы в терминах ее компонентов. В шестидесятые годы информационная система состояла из ряда автономных приложений. Для каждого из этих приложений необходимо было разработать специфичный для конкретного приложения пользовательский интерфейс и систему баз данных, т.е. Каждое приложение имело свои собственные процедуры взаимодействия с пользователем, хранения и повторной проверки данных. В семидесятых годах данные были вытеснены из приложений. Для этой цели были разработаны системы управления базами данных (СУБД). Благодаря использованию базы данных система управления приложениями была освобождена от бремени управления данными. В восьмидесятых годах то же самое произошло с пользовательскими интерфейсами. Появление систем управления пользовательским интерфейсом (UIMSS) позволило разработчикам приложений вытеснить взаимодействие с пользователем из приложений. По нашему мнению, системы управления рабочими процессами являются следующим шагом в вытеснении универсальной функциональности из приложений. Девяностые годы будут отмечены появлением программного обеспечения для документооборота, позволяющего разработчикам приложений вытеснить бизнес-процедуры из приложений. На рисунке 1 четко показано, что, по сути, система управления документооборотом представляет собой общий строительный блок для поддержки бизнес-процессов. Многие информационные системы

такой строительный блок мог бы принести пользу, поскольку многие организации начинают видеть потребность в передовых инструментах для поддержки проектирования и выполнения бизнес-процессов. Существует несколько причин повышенного интереса к бизнес-процессам. Во-первых, общие соображения управления, таких как бизнес-процесс Reengineering (BPR) и непрерывного совершенствования (ИПЦ) стимулировали организацию, чтобы стать более осведомленными бизнес-процессами. Во-вторых, сегодняшним организациям необходимо предоставлять широкий спектр продуктов и услуг. В результате увеличилось количество процессов внутри организаций. Возьмем, к примеру, ипотеку. Десять лет назад существовало всего несколько типов ипотечных кредитов, на данный момент доступно множество типов. За последние три десятилетия увеличилось не только количество продуктов и услуг, но и срок службы продуктов и услуг сократился. В результате современные бизнес-процессы также подвержены частым изменениям. Более того, с течением времени сложность этих процессов значительно возросла. Все эти изменения в среде информационной системы в обычной организации сделали бизнес-процессы важной проблемой при разработке информационных систем. Следовательно, существует явная потребность в строительном блоке под названием "система управления рабочими процессами".

Основной целью бизнес-процесса системы управления является поддержка деф-сайт initiation, выполнение, регистрация и контроль процессов. Поскольку процессы являются доминирующим фактором в управлении документооборотом, важно использовать устоявшуюся структуру для моделирования и анализа процессов документооборота [HL91, Kou95, Law97]. В этой статье мы используем фреймворк, основанный на сетях Петри. Сети Петри являются хорошо обоснованным методом моделирования процессов. Классическая сеть Петри была изобретена Карлом Адамом Петри в шестидесятых годах ([Pet62]). С тех пор сети Петри используются для моделирования и анализа всех видов процессов с различными приложениями, начиная от протоколов, аппаратного обеспечения и встроенных систем и заканчивая гибкими производственными системами, взаимодействием с пользователем и бизнес-процессами. За последние два десятилетия классическая сеть Петри была расширена цветом, временем и иерархией ([Aal94, Jen96]). Эти расширения облегчают моделирование сложных процессов, где важными факторами являются данные и время. Существует несколько причин использования сетей Петри для моделирования рабочих процессов:

#### *формальная семантика*

Рабочий процесс, заданный в терминах сети Петри, имеет четкое определение, поскольку семантика классической сети Петри и несколько усовершенствований (цвет, время, иерархия) были определены формально. *графический характер* Сети Петри - это графический язык. В результате сети Петри интуитивно понятны и просты в освоении. Графический характер также поддерживает связь с конечными пользователями.

### *выразительность*

Сети Петри поддерживают все примитивы, необходимые для моделирования рабочего процесса. Все конструкции маршрутизации, присущие в современных системах управления рабочими процессами, могут быть смоделированы. Более того, тот факт, что состояния представлены явно, позволяет моделировать контрольные точки и неявный выбор. *свойства*

За последние три десятилетия многие люди исследовали основные свойства сетей Петри. Прочная математическая основа позволяет рассуждать об этих свойствах. В результате существует множество общих знаний в виде книг и статей об этой технике моделирования.

### *анализ*

Сети Петри отличаются доступностью многих методов анализа. Очевидно, что это большое преимущество в пользу использования сетей Петри для моделирования рабочих процессов. Эти методы могут быть использованы для доказательства свойства (безопасность свойствами, "инвари" - твии свойства, взаимоблокировки и т. д.) и для расчета показателей результативности деятельности (ре-ответа раза, время ожидания, род занятий и др.). Таким образом, возмож- BLE для того чтобы оценить альтернативные рабочие процессы, используя стандартные Петри-продажи на основе анализа инструменты. *независимый от поставщика*

Сети Петри предоставляют независимую от инструментов платформу для моделирования и анализа процессов. Сети Петри не основаны на программном пакете определенного поставщика и не прекращают свое существование при выпуске новой версии или при замене одного поставщика другим поставщиком.

Другими ссылками, в которых рекомендуется использовать сети Петри для моделирования рабочих процессов, являются [Aal96b, WR96, MEM94, EKR95, EN93, AH97]. В оставшейся части этой статьи мы покажем, как сети Петри могут быть применены к области управления рабочим процессом. Для этого мы сначала познакомимся с основными концепциями управления рабочими процессами и системами управления рабочими процессами. Затем мы введем базовую терминологию сети Петри . В разделе 4 мы покажем, как концепции управления рабочими процессами могут быть отображены на сети Петри. Раздел 5 посвящен анализу процессов документооборота, определенных в терминах сети Петри. Прежде чем мы завершим, мы опишем ряд рабочих инструментов на основе сети Петри, чтобы проиллюстрировать использование сетей Петри в этой области.

## 2 Управление рабочими процессами (системы)

Термин "Управление рабочими процессами" относится к области, которая фокусируется на логике бизнес-процессов. Есть также люди, которые используют этот термин *офисная логистика*. Конечная цель управления рабочим процессом - убедиться, что надлежащие действия выполняются нужным человеком в нужное время. Хотя управление рабочими процессами возможно без использования системы управления рабочими процессами, большинство людей связывают управление рабочими процессами с системами управления рабочими процессами *систем*.

*Коалиция по управлению рабочими процессами* (WfMC) определяет систему управления рабочими процессами следующим образом ([WFM96])..... (tem): *Система, которая полностью определяет, управляет и управляет рабочими процессами посредством выполнения программного обеспечения, порядок выполнения которого определяется компьютерным представлением логики рабочего процесса.* Другие термины для characterize системы документооборота являются: *бизнес-операционной системы*, "рабочая подача управляющего", "менеджер" и "логистические системы управления". С одной стороны, жаль, что управление рабочими процессами часто ассоциируется с системами управления рабочими процессами, потому что это ненужным образом ограничивает область применения управления рабочими процессами. (Управление рабочими процессами возможно осуществлять без использования системы управления рабочими процессами.). С другой стороны, процесс управления системы конкретизировать Основные понятия, приемы и методы для управления бизнес-процессами.

Рабочие процессы чехол на основе, т.е. каждая часть работы выполняется для конкретного случая чехол. Примерами случаев являются ипотека, страховое требование, налоговая декларация, заказ, или запрос информации. Обращения часто генерируются внешним заказчиком. Однако также возможно, что обращение генерируется другим отделом внутри той же организации (внутренний заказчик). Целью управления рабочим процессом является максимально эффективное рассмотрение обращений. Рабочий процесс предназначен для обработки похожих обращений. Обращения обрабатываются путем выполнения задач в определенном порядке. В *Определении процесса workflow* указывается, какие задачи необходимо выполнить и в каком порядке. Альтернативными терминами для определения процесса workflow являются: "процедура", "блок-схема" и "определение маршрута". Поскольку задачи выполняются в определенном порядке, полезно определить условия которые соответствуют причинно-следственным зависимостям между задачами. Условие выполняется или не выполняется (true или false). Каждая задача имеет предварительные и поступления: предварительные условия должны выполняться до того, как задача будет выполнена, а поступления должны выполняться после выполнения задачи. Многие обращения можно обрабатывать, следя одному и тому же определению рабочего процесса. В результате для многих обращений должна выполняться одна и та же задача. Задача, которая должна быть выполнена для конкретного дела называется *рабочий элемент*. Пример работы предмет: выполнение задач 'отправить возврат форма для заказчика на случай жалоба отправлена клиентом пекар'. Большинство рабочих элементов выполнены ресурс. Ресурс - это либо машина (например, принтер или факс), либо человек (участник, рабочий, служащий). В большинстве офисов ресурсы

в основном это люди. Однако, поскольку управление рабочими процессами не ограничивается офисами, мы предпочитаем термин "ресурсы". Ресурсы разрешены для обработки определенных рабочих элементов. Чтобы облегчить распределение рабочих элементов по ресурсам, ресурсы сгруппированы по классам. А *класс ресурсов* - это группа ресурсов со схожими характеристиками.. В одном классе может быть много ресурсов, и ресурс может быть членом нескольких классов ресурсов. Если класс ресурсов основан на возможностях (т.е. функциональных требованиях) его членов, он называется *Роль*. Если классификация основана на структуре организации, такой класс ресурсов называется *организационной единицей* (например, командой, филиалом или отделом). Рабочий элемент, который является быт- рый выполняется определенный ресурс называется *Активность*. Если мы сфотографируем рабочий процесс, мы увидим обращения, рабочие элементы и действия. Рабочие элементы связывают обращения и задачи. Действия связывают обращения, задачи и ресурсы.

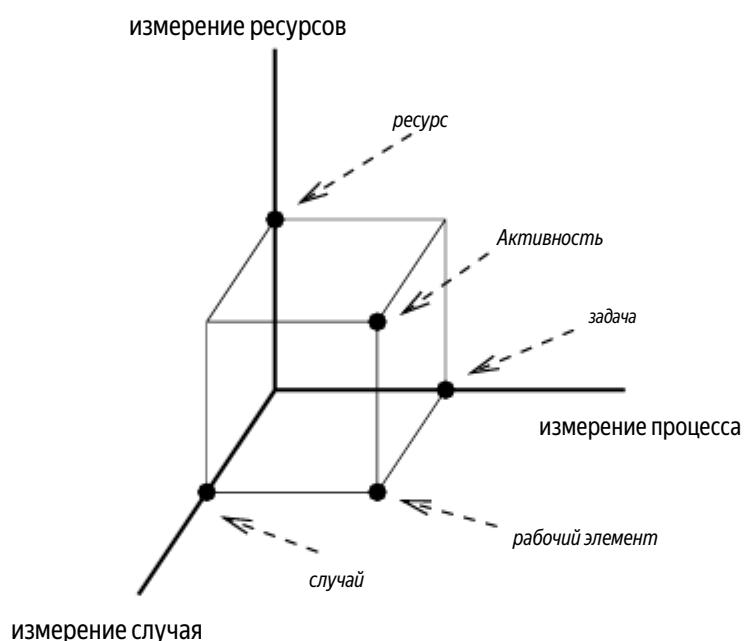


Рисунок 2: Трехмерное представление рабочего процесса.

На рисунке 2 показано, что рабочий процесс имеет три измерения: (1) измерение обращения, (2) измерение процесса и (3) измерение ресурсов. Измерение обращения означает тот факт, что все обращения обрабатываются индивидуально. С точки зрения рабочего процесса обращения напрямую не влияют друг на друга. Очевидно, что они влияют друг на друга косвенно через обмен ресурсами и данными. В измерении процесса задается рабочий процесс, т.е. задачи и маршрутизация по этим задачам. В измерении ресурсов ресурсы сгруппированы по ролям и организационным подразделениям. Мы можем визуализировать рабочий процесс в виде ряда точек в трехмерном представлении, показанном на рис. 2. Каждая точка представляет либо рабочий элемент (обращение + задача), либо действие (обращение + задача + ресурс). На рисунке 2 показано, что управление рабочими процессами является связующим звеном между делами, задачами и организацией.

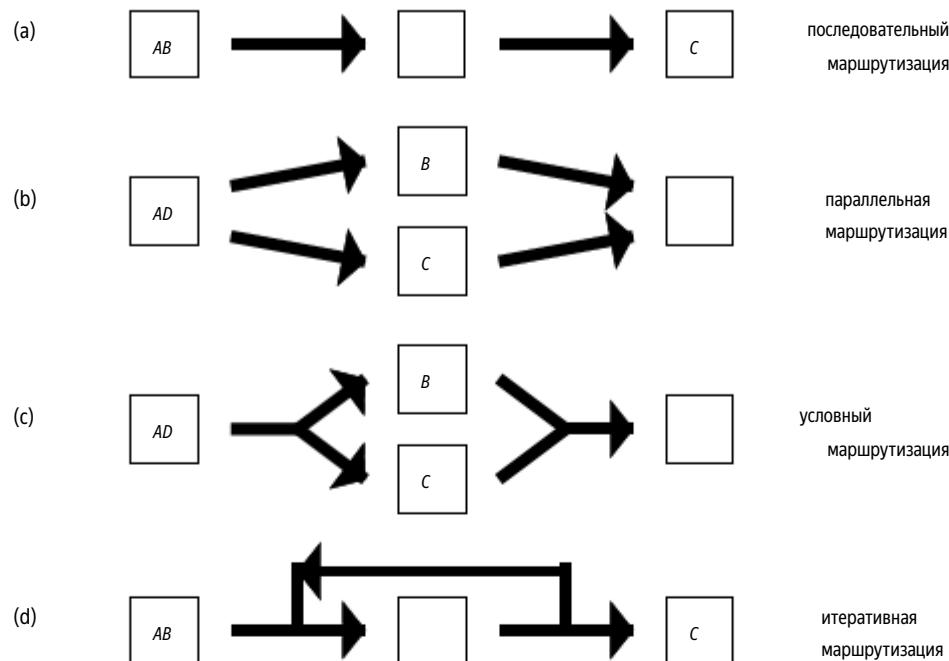


Рисунок 3: Четыре конструкции маршрутизации.

В этой статье мы сосредоточимся на первых двух измерениях, т.е. мы концентрируемся на рабочем потоковом процессе, который определен для обработки обращений. Мы не будем подробно обсуждать аспекты управления персоналом, такие как классификация ресурсов (организационное моделирование) и привязка ресурсов к рабочим элементам (планирование). Мы просто покажем механизмы. Подробное обсуждение управления ресурсами выходит за рамки этой статьи, поскольку мы концентрируемся на применении сетей Петри. Поскольку сети Петри являются методом моделирования процессов, их применение ограничено первыми двумя измерениями.

Для первых двух измерений маршрутизация обращений является одной из основных проблем. Определение процесса рабочего процесса определяет, как обращения распределяются по задачам, которые необходимо выполнить. На рисунке 3 показаны конструкции маршрутизации, определенные Коалицией по управлению рабочими потоками (WfMC). WfMC - это международная организация, миссия которой заключается в продвижении документооборота и установлении стандартов для систем управления документооборотом. WfMC была основана в 1993 году, а в январе 1995 года WfMC выпустила глоссарий, который содержит общий набор терминов для поставщиков рабочих процессов, конечных пользователей, разработчиков и исследователей ([WFM96]). В этом глоссарии определены четыре типа маршрутизации:

#### *последовательные*

Задачи выполняются последовательно, если за выполнением одной задачи следует следующая задача. На рисунке 3(а) задача B выполняется после выполнения задачи A был завершен и перед заданием С начато.

### *параллельный*

На рисунке 3(б) задача *B* и задача *C* выполняются параллельно. Это означает, что *B* и *C* выполняются одновременно или в любом порядке. Для моделирования параллельного маршрутизации определены два строительных блока: (1) соединение *I-split* и (2) соединение *I-*. Разделение "И" на рисунке 3(б) позволяет *B* и *C* будет казнен после *A* было завершено. Функция AND-join синхронизирует два параллельных потока, т.е. задачу *D* может начаться после того, как *B* и *C* будут завершены.

### *условно*

На рисунке 3(с) показана любая из задач *B* или задача *C* выполняется (исключительное ИЛИ). Для моделирования выбора между двумя или более альтернативами мы используем два строительных блока: (1) *ИЛИ-разделить* и (2) *ИЛИ-объединить*. Если задача *A* выполняется, делается выбор между *B* и *C*. Задача *D* может начаться после завершения работы *B or C*.

### *итерация*

Иногда бывает необходимо выполнить задачу несколько раз. На рисунке 3(д) задача *B* выполняется один или несколько раз.

В разделе 4 эти конструкции маршрутизации отображаются на сети Петри. Таким образом, такие понятия, как обращение, задача, рабочий элемент, действие и рабочий процесс документооборота, определяются гораздо более четко. Проблема с запуском также обсуждается в разделе 4. Концепция запуска очень важна в контексте управления рабочими процессами ([Joo94]).

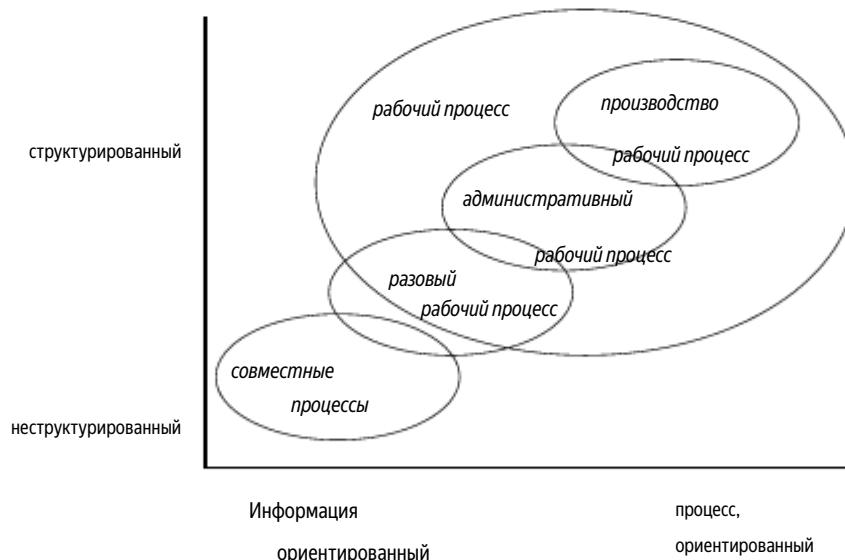


Рисунок 4: Рабочие процессы в сравнении с процессами совместной работы.

Не каждый бизнес-процесс является документооборотом. По нашему мнению, профессионал в области документооборота-

cess характеризуется тремя качествами. Прежде всего, рабочий процесс зависит от конкретного случая . Во-вторых, важным считается сам процесс. В-третьих, процесс можно четко определить. Многие люди используют термин "управление рабочим процессом" в более широком смысле. Например, иногда программные средства groupware, такие как Lotus Notes и Microsoft Exchange, называются системами управления рабочими процессами . Это неверно, поскольку эти продукты не поддерживают сам процесс документооборота, они просто позволяют людям сотрудничать, отправляя сообщения и обмениваясь информацией. На рисунке 4 показан спектр бизнес-процессов, основанных на конкретных примерах. *Производственный процесс production workflow* касается высокоструктурированных процессов практически без вариаций. При таком типе рабочего процесса необходимо обрабатывать большое количество дел в день. Процесс обработки страховых требований является типичным примером производственного процесса. *Административный рабочий процесс.* соответствует процессам, управляемым обращениями, которые следуют четко определенной процедуре. Возможен альтернативный маршрут обращения, но его необходимо предопределить. *Ad-hoc workflow* относится к процессам, в которых процедура не определена (полностью) заранее. В каждом конкретном случае необходимо определить процедуру или изменить существующую процедуру . *Совместные процессы* выходят за рамки нашего определения рабочего процесса. В процессе совместной работы акцент делается на коммуникации и обмене информацией, а не на определении процессов. Для совместных процессов невозможно или не обязательно делать рабочие процессы явными . Современные системы управления рабочими процессами, такие как COSA (COSA Solutions), Flowmark (IBM), OPEN/Workflow (Eastman Software), Staffware (Staffware) и Visual Workflow (FileNet), поддерживают производственный/административный документооборот. Всего несколько продуктов поддерживают нерегламентированный рабочий процесс, например, Ensemble (FileNet). Процессы совместной работы могут поддерживаться средствами группового программного обеспечения, такими как Lotus Notes и Microsoft Exchange. Хотя эти групповые программные инструменты не поддерживают логистический контроль над рабочими процессами, их можно использовать в качестве коммуникационного уровня в проприетарной системе документооборота.

В этой статье мы ограничимся реальными системами управления рабочими процессами, т.е. системами поддержки производственного, административного и / или нерегламентированного рабочего процесса. WfMC также фокусируется на этом типе программных средств. Многие поставщики и пользователи этих реальных систем управления рабочими процессами присоединились к WfMC, чтобы определить общие характеристики этих инструментов, стандартизировать терминологию и определить точные стандартные архитектуры и интерфейсы. Одним из первых результатов, достигнутых WfMC, было определение эталонной модели для архитектуры системы документооборота . Однако, прежде чем мы опишем эталонную модель, мы остановимся на тонком различии между терминами "система управления рабочими процессами" и "система рабочих потоков". Система управления рабочими процессами - это универсальный программный продукт, который может применяться во многих организациях, например, Staffware или COSA. Однако, если система управления рабочими процессами не установлена, не настроена и не заполнена данными об определениях процессов и приложениях, ее нельзя использовать. Поэтому мы оставляем за собой право

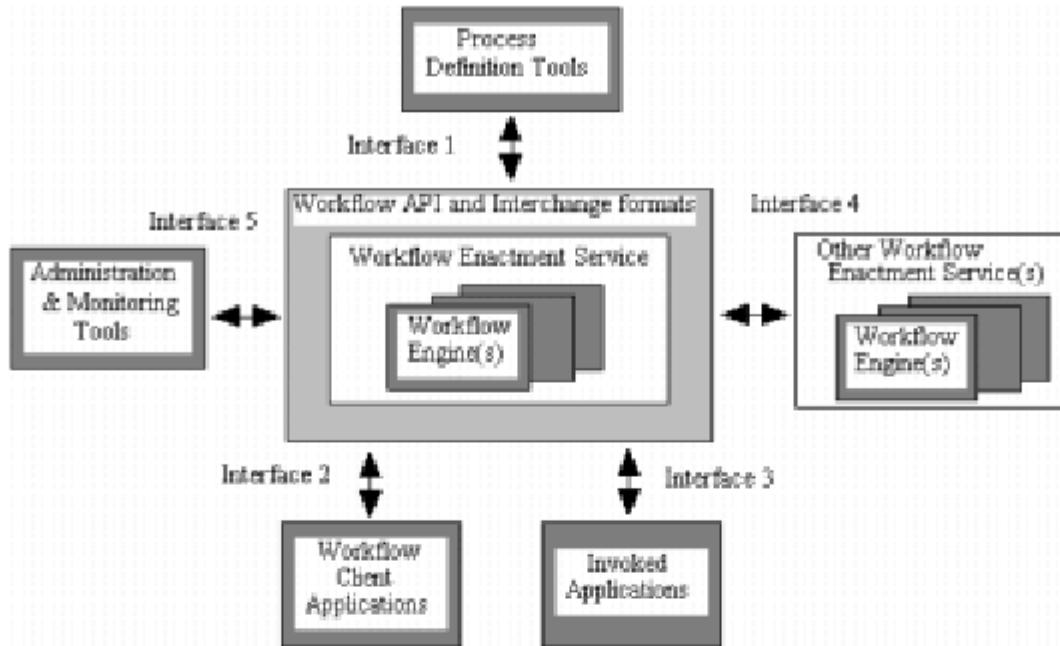


Рисунок 5: Эталонная модель Коалиции по управлению рабочими процессами (WfMC).

термин система рабочих процессов для обозначения всей установленной системы управления рабочими процессами, данных определения процесса, организационных данных, приложений,

прикладные данные, система управления базами данных, файлы конфигурации и другие

программные компоненты находятся на периферии реальной системы управления рабочими процессами.

На рисунке 5 показан обзор этой эталонной модели. Эталонная модель описывает

основные компоненты и интерфейсы в архитектуре рабочего процесса.

Ядром любой системы рабочего процесса является *служба внедрения workflow-процесса*. Служба внедрения workflow предоставляет среду выполнения, которая заботится о контроле

и выполнении workflow-процесса. По техническим или управленческим причинам

служба внедрения рабочего процесса может использовать несколько движков рабочего процесса.

Движок рабочего процесса обрабатывает выбранные части рабочего процесса и управляет выбранными частями

ресурсов. Инструменты определения процессов используются для определения и анализа рабочего процесса,

определения процессов и / или классификации ресурсов. Эти инструменты используются во время проектирования

. В большинстве случаев инструменты определения процесса также можно использовать в качестве набора

инструментов BPR. Большинство систем управления рабочими процессами предоставляют три инструмента определения

процессов: (1) инструмент с графическим интерфейсом для определения процессов рабочего процесса, (2)

инструмент для определения классов ресурсов (модель организации) и (3) инструмент моделирования для анализа

указанного рабочего процесса. Конечный пользователь взаимодействует с документооборотом через интернет

*клиентские приложения рабочего процесса*. Пример рабочего процесса клиентского приложения является

хорошо известной в корзину. Через такую корзину конечному пользователю предлагаются рабочие элементы

. Выбрав рабочий элемент, пользователь может выполнить задачу для конкретного случая. Если

необходимо, на рабочий процесс двигателя вызывает приложений через интерфейс 3. В админис-

инструменты отслеживания используются для мониторинга рабочего процесса и управления им. Эти инструменты используются для регистрации хода рассмотрения обращений и выявления узких мест. Более того, эти инструменты используются для установки параметров, распределения людей и обработки отклонений. Через интерфейс 4 система документооборота может быть подключена к другим системам документооборота. Для стандартизации пяти интерфейсов, показанных на рисунке 5, WfMC стремится к созданию общего *Интерфейс прикладного программирования Workflow (WAPI)*. WAPI задуман как общий набор вызовов API и связанных форматов обмена, которые могут быть сгруппированы вместе для поддержки каждого из пяти интерфейсов (ср. [Law97]).

Архитектура, показанная на рисунке 5, была принята большинством поставщиков. Для большинства интерфейсов были предложены стандарты. К сожалению, предлагаемые стандарты находятся на техническом уровне с акцентом на синтаксис, а не на семантику. На концептуальном уровне консенсуса нет. Рассмотрим, например, интерфейс 1. Синтаксис формата обмена был определен без формализации значения состояний и основных строительных блоков, таких как И/ИЛИ-split/join. Следовательно, существует потребность в концептуальном стандарте, который четко определяет семантику необходимых базовых конструкций. Мы утверждаем, что структура рабочего процесса, основанная на сетях Петри, была бы хорошим кандидатом для такого стандарта. В оставшейся части этой статьи мы обосновываем это утверждение.

### 3 Сети Петри

В данном разделе представлены базовые сети Петри терминологии и нотации. Читатели, Альфа- милиар с сетями Петри можете пропустить этот раздел.<sup>1</sup> Исторически сложилось так, что сети Петри берут свое начало в ранних работах Карла Адама Петри ([Pet62]). С тех пор использование и изучение сетей Петри значительно расширились. За обзором истории сетей Петри и обширной библиографией читатель может обратиться к [Mur89].

#### 3.1 Классическая сеть Петри

Классическая сеть Петри представляет собой ориентированный двудольный граф с двумя типами узлов, называемыми *places* и *transitions*. Узлы соединены направленными *дугами*. Соединения между двумя узлами одного типа не допускаются. Места обозначены кружками, а переходы - прямоугольниками.

---

<sup>1</sup> Обратите внимание, что состояния представлены взвешенными суммами, и обратите внимание на определение (элементарных) путей.

**Определение 1 (сеть Петри)** Сеть Петри - это тройная

$P(TF)$

- $\Pi$  является конечным набором мест,
- $T$  является конечным набором переходов ( $P \setminus T = \emptyset$ ),
- $\Phi = P \cup T_1 \cup T_2 \cup P$  представляет собой набор дуг (отношение потока)

Место называется **входным местом** перехода, если существует направленная дуга из

р в место  $t$ . Место называется **моментом выхода** перехода, если существует направленная дуга

от  $t$ . Мы используем  $t$  для обозначения набора входных мест для перехода. Примечание-

связи  $p$ ,  $t$  и  $r$  имеют схожие значения, например, является ли набор  $P$  переходов общим

$p$  в качестве места ввода. Обратите внимание, что мы ограничиваемся дугами с

весом 1. В контексте процедур рабочего процесса нет смысла указывать другие

веса, потому что места соответствуют условиям. В любой момент место

содержит ноль или больше токенов привлеченный в качестве черных точек. В **состояние**,

часто называемая маркировкой, это распределение жетонов по местам, т.е.,

$M_2$

$P = !$  В. Мы представим состояние следующим образом:

$1p_1 \quad 2p_{+2} \quad 3p_{+1} \quad 4p_{+0}$

является ли состояние

$p$  с одним жетоном на месте 1, два токена в 2, один токен в 3 и никаких жетонов в 4. Мы также можем

$p$

представить это состояние следующим образом:  $1p_2 P$ : частичный  $M$  порядок. Для двух состояний  $M$  diff для всех

$1M_1 \quad 2M_2$

$p$

Количество токенов может меняться во время выполнения сети.

Переходы являются активными компонентами сети Петри: они изменяют состояние сети в соответствии со следующим **правилом срабатывания**:

(1) Считается, что переход **включен** эквивалентность каждого входного места  
один жетон.

$p t$  содержит по крайней мере

(2) Включенный переход может **сработать**. Если переход сработает, то **потребляет** от одного до  
кен из каждого места ввода и **выдает** по  $p$   $t$   
одному токену в каждом месте вывода  $p$

Учитывая сеть Петри  $P(TF)$  и состояние  $M_1$ , у нас есть следующие обозначения:

- $1M_1 \quad !_t$  переход разрешен в состоянии  $M_1$  и увольнение  $t$  в  $M_1$  приводит к состоянию  $M_2$
- $1M_1 \quad !_t \quad 2M_2$  существует такой переход, что  $t = T_1 T_2 \dots T_n$ ;  $M_1$  ведет из штата  $M_2$  для
- $1M_1 \quad !_n M_n$ : последовательность обжига состояния  $M_n$ , т.е.,  $1M_1 \quad !^t \quad 2M_2 \quad !^{t_2} \quad \dots \quad !^{t_n} \quad M_n$

$M_n$  Последовательность достижимым От состояний  $ing = t_1 \cdot T \cdots T n$ ;  $t_1$  такой, что последовательность также разрешена, т.е.,

$$1M \quad (обозначение 1MM ! M), \text{ если там есть ель-} \\ 1M ! \quad .n \text{ Обратите внимание, что холостая стрельба} \\ 1M ! \quad 1M$$

( $PN$ ) Мы используем  $PN$  обозначают сеть Петри  $PN$  с начальным состоянием  $M$ . Состояние  $M^0$  - это достижимое состояние из  $M_j$  iff  $M ! M^0$ . Давайте определим некоторые свойства для сетей Петри

**Определение 2 (текущее) Сеть Петри** ( $PN$ )  $M_j$  и у каждого перехода есть состояния  $M^{00}$  достижимое из  $M^0$  достижимого состояния, которое позволяет  $M^0$  является текущим iff для каждого

**Определение 3 (Ограниченнная, безопасная) Сеть Петри** ( $PN$ )  $M_j$  ограничена, если для каждого места  $p$  существует натуральное число  $n$  такой, что для каждого достижимого состояния количество  $p$  меньше. Сеть безопасна, если для каждого места максимальное количество токенов в  $of tokens$  не превышает 1.

Пути соединяют узлы последовательностью дуг. Путь элементарен, если каждый узел уникален. Для удобства мы также вводим оператор алфавита для путей.

**Определение 4 (Путь, элементарный, алфавит)** Пусть  $PN$  - сеть Петри. Дуга  $h$  такой, что  $h : n_i \rightarrow n_j$  ибо  $h$  это алфавит из. элементарно, если для любых двух узлов  $n_i$  и  $n_j$   $n_i h \cdots n_j$  на  $C = jk$

**Определение 5 (Сильно связанный)** Сеть Петри является сильно связанный, если для каждого пары узлов (т.е. мест и переходов)  $x$  и  $y$  есть путь, ведущий от  $x$  для.

### 3.2 Высокоуровневые сети Петри

Классическая сеть Петри позволяет моделировать состояния, события, кондиции, синхронизацию, параллелизм, выбор и итерацию. Однако сети Петри, описывающие реальные процессы, как правило, сложны и чрезвычайно велики. Более того, классическая сеть Петри не позволяет моделировать данные и время. Для решения этих проблем было предложено множество расширений. Тремя хорошо известными расширениями базовой модели сети Петри являются: (1) расширение с использованием цвета для моделирования данных, (2) расширение со временем и (3) расширение с иерархией для структурирования больших моделей.

Сеть Петри, расширенная цветом, временем и иерархией, называется *высокоуровневой сетью Петри*. В этой статье мы будем использовать эти расширения для моделирования конкретных аспектов. Однако формализация этих аспектов выходит за рамки данной статьи. Поэтому мы вводим эти три расширения в неформальной манере.

#### *расширение с помощью цвета*

Токены часто представляют объекты (например, ресурсы, товары, людей) в модифицированной системе. Поэтому мы часто хотим представить атрибуты этих объектов. Если страховое требование моделируется с помощью токена в сети Петри, мы хотим представлять такие атрибуты, как имя заявителя, идентификационный номер, дату и сумму. Поскольку эти атрибуты нелегко представить с помощью маркера в классической сети Петри, мы расширим модель сети Петри с помощью *цветных* или *типализированных токенов*. В цветной сети Петри каждый токен имеет значение, часто называемое "цветом". Переходы определяют значения произведенных токенов на основе значений потребленных токенов, т. е. переход описывает отношение между значениями "входных токенов" и значениями "выходных токенов". Также возможно указать "предварительные условия", которые учитывают цвета потребляемых токенов.

#### *расширение со временем*

Для реальных систем часто важно описать *временное поведение* системы, т.е. Нам нужно смоделировать длительности и задержки. Поскольку классическая сеть Петри не способна обрабатывать количественное время, добавлена концепция синхронизации. Существует множество способов ввести время в сеть Петри. Время может быть связано с символами, местами и / или переходами.

#### *расширение с иерархией*

Хотя временные цветные сетки Петри позволяют кратко описать многие бизнес-процессы, точные спецификации для реальных систем имеют тенденцию становиться большими и сложными. Именно по этой причине мы предоставляем иерархическую конструкцию, называемую *подсеть*. Подсеть представляет собой совокупность нескольких мест, переходов и подсистем. Такую конструкцию можно использовать для структурирования больших процессов. На одном уровне мы хотим дать простое описание процесса (без необходимости рассматривать все детали). На другом уровне мы хотим описать более подробное поведение. Расширение с иерархией допускает такой подход.

Для более подробного обсуждения этих расширений и других видов высокогородневых сетей Петри читателю предлагается обратиться к [Jen96, Aal94, Hee94].

## **Отображение концепций управления рабочими процессами на сети Петри**

Большинство систем и методологий управления рабочими процессами для поддержки управления рабочими процессами отделяют моделирование процесса документооборота (как?) От моделирования структуры организации и ресурсов внутри организации (кем?). Различие между этими двумя аспектами показано на рисунке 2, который связывает измерение процесса и измерение ресурсов. Существует много причин для разделения этих двух измерений при определении рабочего процесса. Снижается сложность, стимулируется повторное использование, и появляется возможность модифицировать процесс без изменения организационной модели (и наоборот). В этом разделе мы сначала покажем, что сети Петри подходят для моделирования измерения процесса. Затем мы покажем, как включить измерение ресурсов.

### **4.1 Определение процесса рабочего процесса**

В измерении процесса указано, какие задачи необходимо выполнить и в каком порядке. Моделирование определения рабочего процесса в терминах сети Петри довольно простое.: *задачи* моделируются с помощью *переходы*, *условия* моделируются с помощью *places*, а *случаи* моделируются с помощью токенов. Чтобы проиллюстрировать отображение концепций управления рабочими процессами на сети Петри, мы рассмотрим обработку жалоб. Сначала жалоба регистрируется (задача *регистрация*), затем параллельно жалобщику отправляется анкета (задача *отправить вопросaire*) и жалоба оценивается (задача *оценить*). Если заявитель вернет анкету в течение двух недель, задача *обработать анкету* будет выполнена. Если — анкета не возвращается в течение двух недель, результат анкеты аннулируется (время выполнения *задания истекло*). На основании результатов оценки жалоба обрабатывается или нет. Фактическая обработка жалобы (задача *обработать жалобу*) откладывается до тех пор, пока анкета не будет обработана или не наступит тайм-аут. Процесс рассмотрения жалобы проверяется с помощью задачи *обработка проверки*. Наконец, задача *Архив* выполняется.

На рисунке 6 показано определение рабочего процесса для обработки жалоб, заданное в терминах сети Петри. Задачи *Зарегистрироваться, отправить анкету, оценить, про- обработать анкету, время ожидания, обработать жалобу, проверить обработку и Архив* были смоделированы с помощью переходов. Переходы *обрабатываются нормально и NOK обработки добавлены для моделирования двух возможных результатов выполнения задачи проверьте процесс обработки*. Переходы *без обработки и требуется обработка* были добавлены по аналогичным причинам. Для моделирования состояний между задачами были добавлены условия. Каждое условие моделируется местом. Например, место *c2* соответствует условию "готов рассмотреть жалобу". Условие *c5* верно (т.е. соответствует месту *c5* соп-

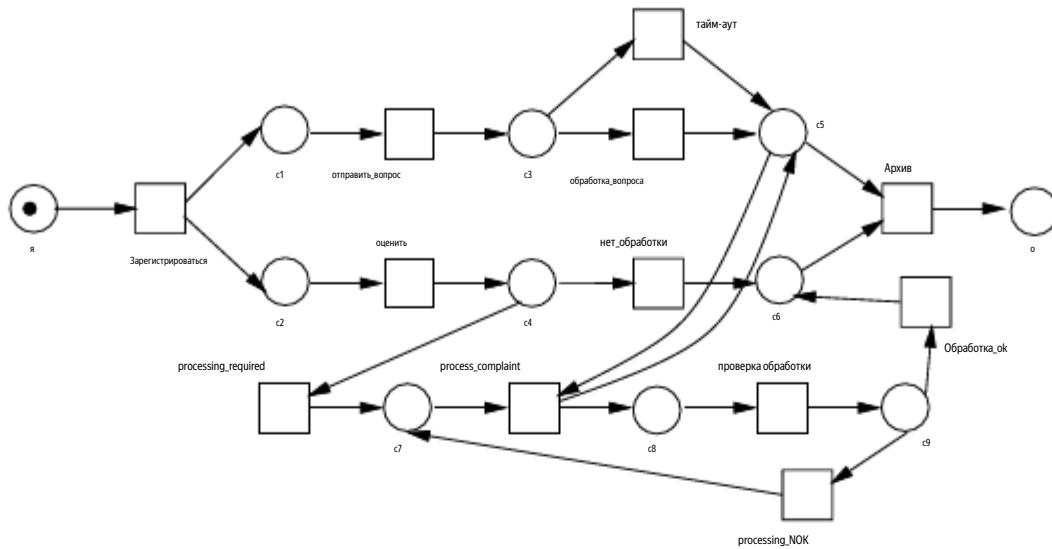


Рисунок 6: Сеть Петри для обработки жалоб.

сохраняет токен), если анкета была обработана или время ожидания истекло.

Обратите внимание, что  $c_5$  является обязательным условием для выполнения задачи *Архив* и задача *обработать жалобу*. –

Условие *я* является начальным условием и condition *о* это конечное условие.

Определение процесса workflow, показанное на рисунке 6, моделирует жизненный цикл одного случая. В целом, существует множество случаев, которые обрабатываются в соответствии с одним и тем же определением процесса workflow. Каждый из этих случаев соответствует одному или нескольким tokens. Если токены нескольких случаев находятся в одной сети Петри, то эти токены могут смешиваться. Например, переход *Архив* может использовать два токена, которые соответствуют разным случаям. Очевидно, что это нежелательно. Есть два способа решить эту проблему. Прежде всего, можно использовать высокоровневую сеть Петри, где каждый токен имеет значение (цвет), которое содержит информацию об идентичности соответствующего регистра (case identifier). Переходы не могут запускаться, если регистровые идентификаторы используемых токенов не совпадают, т. е. Используется предварительное условие, которое проверяет и сравнивает значения токенов. На рисунке 7 показана высокоровневая сеть Петри с идентификаторами случаев. Пунктирные линии используются для привязки маркеров к случаям, например, маркер на месте  $c_1$  имеет значение 3, что связывает его со случаем 3. Другой способ решить эту проблему заключается в следующем. Каждому случаю соответствует уникальный экземпляр

Сеть Петри. Если существуют  $n$  случаи, тогда есть  $n$  экземпляры сети Петри. Можно представить такой экземпляр как слой. Если эти слои наложить друг на друга, то можно увидеть случаи на одной и той же диаграмме. Последнее интересно с управлением точки зрения, потому что можно получить общее представление о состоянии рабочего процесса. Например, если место содержит много токенов, это может указывать на узкое место. В оставшейся части этой статьи мы рассмотрим сети Петри, которые описывают жизненный цикл одного случая изолированно.

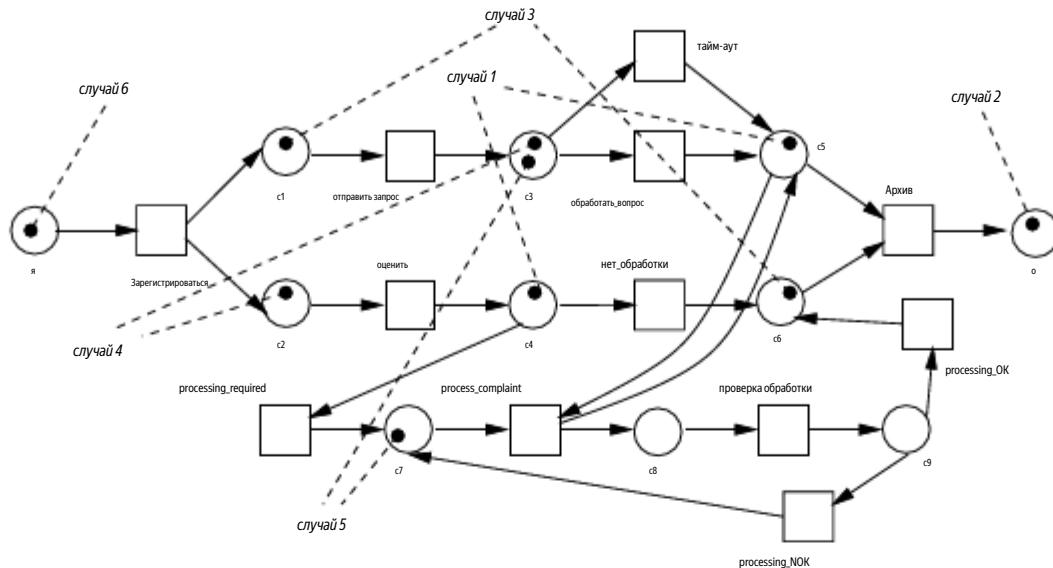


Рисунок 7: Токены имеют идентификатор обращения, который позволяет разделять обращения.

Сеть Петри, которая моделирует определение рабочего процесса (т.е. Жизненный цикл одного обращения в отдельности), называется *Сеть рабочего процесса (WF-net)*. WF-сети удовлетворяют двум требованиям-требованиям. Прежде всего, WF-сеть имеет одно место ввода () и одно место вывода

$я$  ( $y$ ). Токен в  $я$  соответствует обращению, которое необходимо обработать, токен в

$о$  ( $o$ ) соответствует обращению, которое было обработано. Во-вторых, в WF-сети нет висячих задач и / или условий. Каждая задача (переход) и условие (место)

должен способствовать рассмотрению дел. Следовательно, каждый переход (место)

должен располагаться на пути от места  $я$  разместить  $о$ . Последнее требование соответствует-

соответствует сильной связности , если  $о$  подключается через дополнительный переход

$t$  (смотрите рисунок 23).

**Определение 6 (WF-net)** Сеть Петри

$PN = \langle T, F, I, O, M_0 \rangle$

является WF-сетью (WorkFlow net)

тогда и только тогда, когда:

- (i)  $PN$  имеет два особых места:  $я$  . Место - это исходное место:  $я$   $о$  является местом для умножальника:

(ii) Если мы добавим  $f$  переход  $t$  к  $PN$ , который соединяет место  $я$  с помощью (т.е.  $t = f_0 g$ )

$t = я \rightarrow о$  ), тогда результатирующая сеть Петри будет прочно связана.

Места в наборе

$P$

$T$  соответствовать условиям, переходам в

поставленных задачах. Обратите внимание, что требования, указанные в Определении 6, являются минимальными требованиями.

Даже если эти требования удовлетворяются все еще возможно определение бизнес-процесса про-  
цесса определение потенциальных взаимоблокировок и/или livelocks.

Жетоны в ВФ-объем представлять состояние рабочего процесса отдельного обращения. Рабочий процесс

состояние содержит частичную информацию о состоянии обращения. Кроме того, обращение содержит атрибуты рабочего процесса и данные приложения. Состояние рабочего процесса соответствует распределению токенов по местам (разметке). Рассмотрим, например, рисунок 7.

Состояние рабочего процесса в случае 5 равно  $c3 \parallel c7$ . Атрибут рабочего процесса - это определенная часть + информация, используемая для маршрутизации обращения. Атрибут рабочего процесса можно рассматривать как управляющую переменную или логистический параметр. Атрибутом рабочего процесса может быть возраст подателя жалобы, отдел, ответственный за рассмотрение жалобы, или дата регистрации.

Мы также будем использовать термин атрибут обращения для ссылки на атрибут рабочего процесса конкретного случая. Классическая сеть Петри

$PN = (T, F)$

абстрагируется от этих атрибутов

рабочего потока. Однако в высокораспределенной сети Петри расширение с цветом может использоваться для моделирования атрибутов рабочего процесса. Данные приложения не используются для управления рабочим процессом, они используются для выполнения задачи. Данные приложения выходят за рамки определения рабочего процесса, и система управления рабочим процессом не имеет сведений об этой информации. (Тем не менее, часто существуют атрибуты рабочего процесса, которые могут быть получены из данных заявки.) Примерами данных заявки являются адрес заявителя и отчет об оценке.

## 4.2 Конструкции маршрутизации

В измерении процесса строительные блоки, такие как AND-split, AND-join, OR- разделение и OR-join, используются для моделирования последовательной, условной, параллельной и итеративной маршрутизации (WfMC [WFM96]). Очевидно, что WF-сеть может использоваться для указания маршрутизации обращений. В разделе 2 были определены четыре типа маршрутизации: последовательная, параллельная, условная и итерационная.

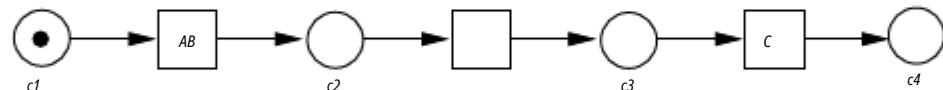


Рисунок 8: Последовательная маршрутизация.

*Последовательная маршрутизация* используется для анализа причинно-следственных связей между задачами. Рассмотрим две задачи A и B. Если задача B выполняется после завершения задачи A, тогда A и B выполняются последовательно. На рисунке 8 показано, что последовательная маршрутизация может быть смоделирована путем добавления мест. Место c2 моделирует причинно-следственную связь между задачами A и задача B, то есть разместить c2 представляет собой постусловие для задачи A и предварительное условие для выполнения задачи B. Место c3 моделирует причинно-следственную связь между задачами B и задача C.

*Параллельная маршрутизация* используется в ситуациях, когда порядок выполнения менее строгий.

Для примера, две задачи B и C необходимо выполнить, но порядок выполнения является произвольным. *bitrary*.

Для моделирования такой параллельной маршрутизации используются два стандартных блока: (1)

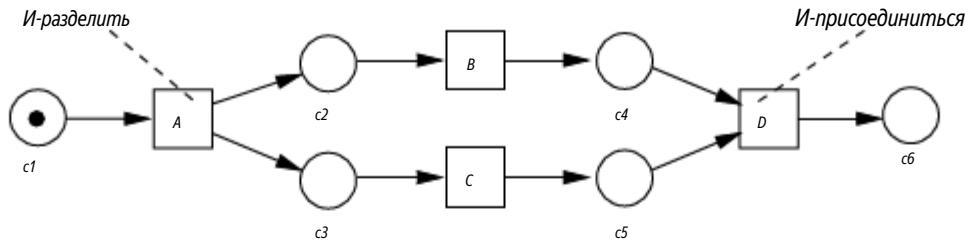


Рисунок 9: Параллельная маршрутизация.

**И-разделить и (2) И-объединить.** На рисунке 9 показано, что оба строительных блока могут быть смоделированы обычными переходами. Эти переходы можно рассматривать как задачи управления которые были добавлены для целей маршрутизации. Однако обычная задача также может действовать как AND-split и / или AND-join . Выполнение AND-split A включает обе задачи B и задача C. И-присоединяйтесь D включается после выполнения обоих B и C, т. е., D используется для синхронизации двух подпотоков. В результате задача B и задача C выполняются параллельно. Это означает, что B и C выполняются в произвольном порядке. Возможно даже, что выполнение B и C совпадает, т.е. для одного случая задачи накладываются друг на друга по времени. В нескольких системах управления рабочими процессами невозможно выполнить две задачи для одного и того же случая одновременно. В данном случае мы используем термин *полупараллельный*. Если на рисунке 9 используется полупараллельная маршрутизация, то B выполняется до C или наоборот, т.е. семантика чередования используется для интерпретации определения рабочего процесса. Иногда полупараллельная маршрутизация используется, чтобы избежать серьезных технических проблем. Например, в каждом конкретном случае приложения и/или задачи в системе управления рабочими процессами совместно используют данные. В этом случае механизм блокировки в системе управления базой данных может препятствовать одновременному выполнению задач для одного и того же случая.

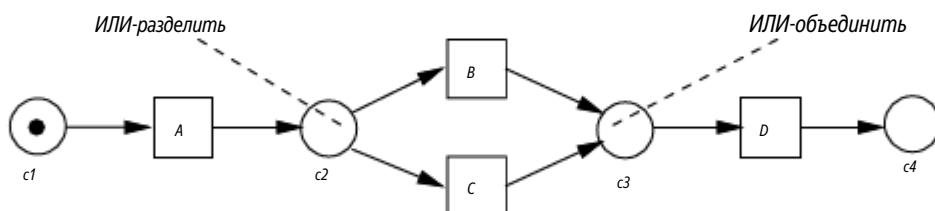


Рисунок 10: Условная маршрутизация.

**Условная маршрутизация** используется для обеспечения маршрутизации, которая может различаться в разных случаях. Таким образом, маршрутизация обращения может зависеть от атрибутов рабочего процесса обращения, поведения среды или рабочей нагрузки организации. Для моделирования выбора между двумя или более альтернативами используются два строительных блока: (1) разделение ИЛИ и (2) объединение ИЛИ (в обоих случаях исключающее ИЛИ). Разделение ИЛИ может моделироваться местом с несколькими исходящими дугами, соединение ИЛИ моделируется

место с несколькими входящими дугами. На рисунке 10 показана ситуация, когда задача A за следует любая из задач B или задача C, т. е. делается выбор между B и C. За выполнением одной из этих двух задач следует выполнение задачи D. Место c2 является предварительным условием для обоих B и C. Однако для данного случая будет выполнена только одна из этих двух задач. с1.

Если c2 содержит токен, делается недетерминированный выбор между B и C. Однако выбор между альтернативами часто зависит от атрибутов рабочего процесса. Если выбор основан на атрибутах рабочего процесса, это детерминированный выбор. Например, направление страхового требования может зависеть от стоимости компенсации. Следовательно, атрибут рабочего процесса используется для учета затрат на компенсацию. Если эти затраты превышают определенную сумму, необходимы дополнительные проверки. Маршрут нарушения правил дорожного движения может зависеть от типа нарушения правил дорожного движения, представленного соответствующим атрибутом рабочего процесса. Есть два способа смоделировать выбор на основе атрибутов рабочего процесса. (Напомним, что расширение с цветом используется для изменения атрибутов рабочего процесса.) Мы можем использовать конструкцию, показанную на рисунке 10, и добавить предварительное условие (т. Е. Дополнительное разрешающее требование, основанное на одном или нескольких атрибутах рабочего потока) к каждой из задач таким образом, чтобы либо B или C включается, если c2 содержит маркер. Другой способ моделирования детерминированного выбора между B или C отображается в

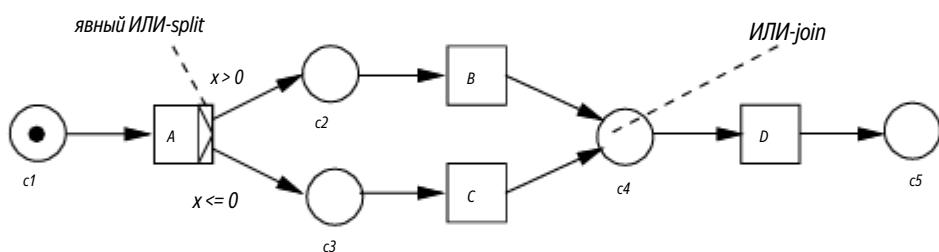


Рисунок 11: Явный выбор между B и C на основе атрибута рабочего процесса x.

Рисунок 11. Переход A имеет два выходных места c2 и c3. Переход A выдает ei-токен в c2 или c3. Выбор между c2 или c3 основан на атрибуте рабочего процесса x. Если x является позитивным, задача B будет выполнено, в противном случае задача C. Специальный символ используется для обозначения того факта , что задача A является ИЛИ-разделенным (эксклюзивным ИЛИ). Обратите внимание, что не- детерминированный выбор на рис. 10, отличается от выбора на рис. 11, в отношении к моменту выбора. На рис. 11 выбор сделан данный момент задач A завершено, на рисунке 10 выбор сделан в тот момент, когда B или C выполняется. Мы будем использовать термин явный Или-split для выбора на основе атрибутов рабочего процесса. Термин неявный Или-разделенный зарезервирован для ситуации, когда момент выбора наступает как можно позже . Для моделирования рабочего процесса крайне важно различать неявное и явное разделение OR.

На рисунке 11 мы ввели специальное обозначение для явного разделения OR. Мы также

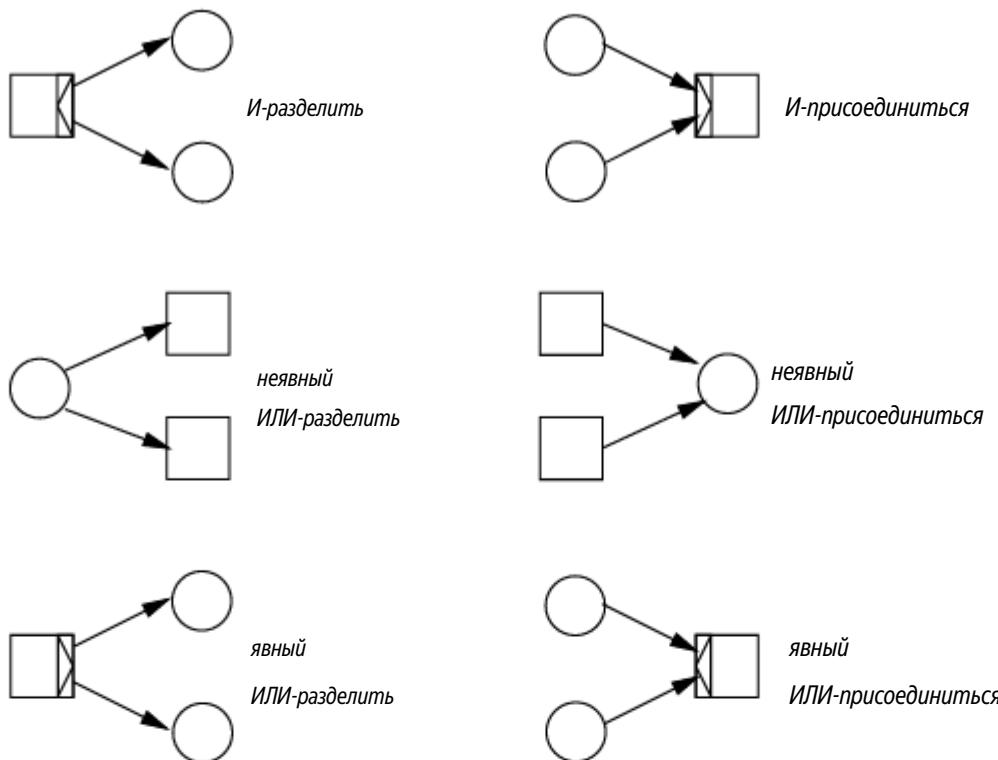


Рисунок 12: Строительные блоки для моделирования рабочего процесса.

обозначьте тот факт, что задача состоит из И-разделения, И-объединения и/или ИЛИ-объединения. На рис. 12 перечислены основные конструкции. И-разделение и И-объединение соответствуют нормальному поведению перехода в классической сети Петри. Неявные OR-split и OR-join моделируются places . Явную или-сплит моделируется с помощью Тран- приобретение, которое производит один жетон на один ее выход находится, т. е. исключительное или (XOR). Место выбирается на основе атрибутов рабочего процесса. Пример предложение ИЛИ объединение моделируется переходом, который включается, если одно из входных мест содержит маркер. В общем, явное ИЛИ-соединение может быть смоделировано неявным ИЛИ-соединением (т. Е. местом). Нет настоятельной необходимости проводить различие между неявным и явным ИЛИ-соединениями. Поэтому в этом документе мы будем избегать использования явных объединений OR. Однако, различие между неявным ИЛИ-разделением и явным ИЛИ- разделением имеет практическое значение и будет обсуждаться более подробно, когда будет введена концепция запуска.

Последняя форма маршрутизации - это *итерация*. Итерацию можно смоделировать, используя строительные блоки, показанные на рисунке 12. Рассмотрим, например, определение рабочего процесса, показанное на рисунке 13. Задача C это контрольная задача, которая проверяет результат выполнения задачи B. Основываясь на этой проверке, задача B может быть выполнен еще раз. На рисунке 13 задача B выполняется один или несколько раз. Также можно указать эту задачу B выполняется ноль или более раз. Итерация часто считается нежелательной формой маршрутизации,

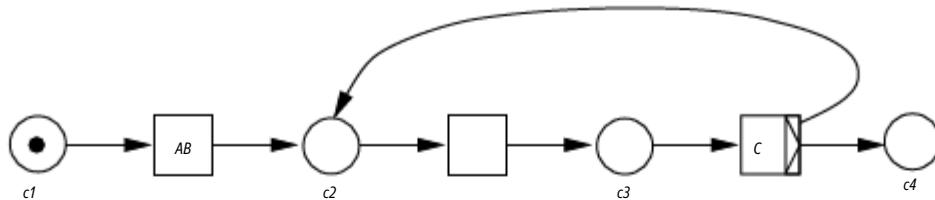


Рисунок 13: Итерация: *B* может выполняться несколько раз.

потому что это соответствует повторному выполнению одной и той же задачи без достижения какого-либо реального прогресса. К сожалению, есть много ситуаций, когда повторения невозможно избежать. Например, информация, предоставленная клиентом, является неполной или в запросе отказано.

### 4.3 Запускающий

Определение процесса workflow, подобное тому, которое показано на рис. 6, определяет, как направляется обращение, т.е. Какие задачи необходимо выполнить и в каком порядке. Для произвольного случая в определенном состоянии (состояние рабочего процесса и атрибуты рабочего процесса) указано, какие задачи *могут* выполняться. Важно отметить, что тот факт, что если задача может быть выполнена для конкретного случая, то это не означает, что задача выполняется напрямую. Например, если задача должна быть выполнена сотрудником, то сотрудник должен быть доступен и желать выполнить эту задачу. Если сотрудник болен, в отпуске или на обеде, то задание выполнено не будет. Другим примером является обработка формы, возвращаемой заказчиком. Если клиент не вернет форму, соответствующая задача не может быть выполнена. Эти примеры показывают, что для многих задач необходимо выполнять дополнительные условия (например, доступность ресурсов или информации). Выполнение этих задач не может быть принудительно выполнено системой управления рабочими процессами. Система управления рабочими процессами не может заставлять клиентов возвращать формы и не может заставлять сотрудников выполнять определенные задачи. Система управления рабочими процессами не осуществляет полный контроль, она лишь поддерживает рабочий процесс. Поэтому важно проводить различие между включением задачи и выполнением задачи. Поскольку включение задачи не подразумевает, что задача будет выполнена (немедленно), крайне важно проводить это различие. Поэтому мы вводим концепцию запуска.

А *Триггер* - это внешнее условие, которое приводит к выполнению разрешенной задачи.

Выполнение экземпляра задачи для конкретного случая начинается в тот момент, когда запускается позиция задачи *in-*. Экземпляр задачи может быть запущен только в том случае, если соответствующий случай находится в состоянии, которое позволяет выполнить задачу. В этой статье мы различаем четыре типа задач.

*Автоматический*: задача запускается в тот момент, когда она включена. Этот вид тригонометрии используется для задач, которые выполняются приложением, которое не выполняет

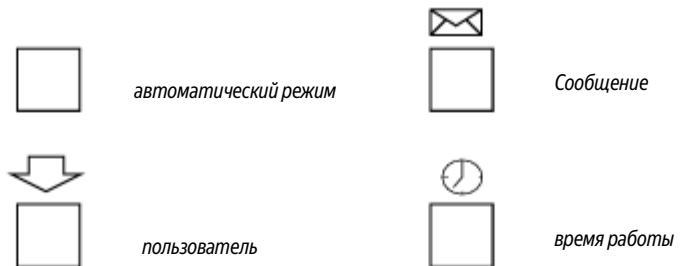


Рисунок 14: Четыре типа запуска.

требуется взаимодействие с человеком.

**Пользователь:** задача запускается участником-человеком, т. Е. Пользователь выбирает включенный экземпляр задачи для выполнения. В системе управления рабочими процессами у каждого пользователя есть так называемая "корзина входящих". Эта корзина содержит экземпляры задач, которые включены и могут выполняться пользователем. При выборе экземпляра задачи запускается соответствующий экземпляр задачи.

**Сообщение:** внешнее событие (т. Е. Сообщение) запускает включенный экземпляр задачи. Примерами сообщений являются телефонные звонки, факсимильные сообщения, электронные письма или EDI сообщения.

**Время:** включенный экземпляр задачи запускается по часам, т.е. задача выполняется в заранее определенное время. Например, задача "удалить документ" запускается, если обращение находится в определенном состоянии более 15 часов.

Только для автоматических задач включение и фактический запуск выполнения совпадают . В определении процесса workflow мы будем использовать символы, показанные на рис. 14, для обозначения типа требуемого триггера.

На рисунке 15 показано определение рабочего процесса для обработки жалоб с использованием стандартных блоков, представленных на рисунке 12. Задачи *обработка* не требуется, обработка не требуется, обработка В порядке, и NOK обработки (см. рисунок 6) заменены – на OR-split evaluate и OR-split check processing. Диаграмма также дополнена информацией о запуске. Задачи *Зарегистрироваться, оценить, завершение процесса нет и проверить обработку* требуется пользовательский триггер, поскольку они выполняются персоналом man resources. Задачи *отправить анкету и Архив* выполняются с помощью приложения, которое не требует взаимодействия с человеком. Задача *обработать анкету* может быть выполнено только в том случае, если заявитель вернет форму, т. Е. Для выполнения этой задачи требуется внешний триггер. Задача *время ожидания* требует временного триггера. Если состояние рабочего потока обращения таково, что в c3 (рисунок 15) присутствует маркер, то есть две возможности: либо *обработать анкету* выполняется, потому что заявитель повторно вовремя сдает форму или *тайм-аут* выполняется по истечении определенного периода времени

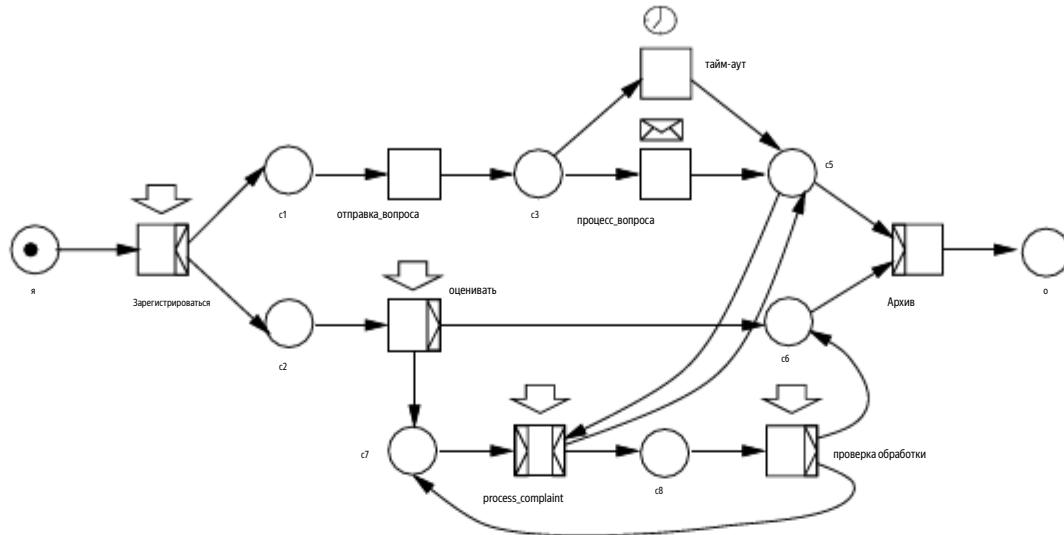


Рисунок 15: Определение рабочего процесса, дополненное информацией о запуске.

истекает срок действия. Обратите внимание, что момент выбора между *опросником* процесса и *тайм-аутом* выполняется как можно позже. Следовательно, вместо явного используется неявное ИЛИ-split ИЛИ-split.

Введение триггеров может быть использовано для иллюстрации сути различия между неявным и явным разделением OR. Если неявное ИЛИ-split на рисунке 15 заменить на явное ИЛИ-split (т.е. *отправить анкету* выбирает между *процессным листом* и *тайм-аутом*), поведение резко меняется. Например, возможно, что формы, возвращенные вовремя, будут отклонены, или обращения будут заблокированы, если некоторые формы не будут возвращены вообще. Другой пример приведен на рисунке 16. В обоих определениях процесса выполнение задачи A за этим следует выполнение B или C. В первом определении процесса (a) момент выбора наступает настолько поздно, насколько это возможно. После выполнения A, существует "гонка" между B и C. Если внешнее сообщение требуется для выполнения задачи C прибывает до того, как кто-то приступает к выполнению задачи B, тогда C выполняется, в противном случае B. Во втором определении процесса рабочего процесса (b) выбор фиксируется после выполнения A. Если задача B если выбрано значение, то приход внешнего окончательного сообщения не влияет. Если C выбирается, затем B не может быть использован для обхода C. На рисунке 16 показана важность явного представления триггеров, состояний и момента выбора. Многие системы управления рабочими процессами абстрагируются от состояний между последующими задачами, поэтому у них возникают проблемы с моделированием контрольных этапов таких как c5 на рисунке 15. Очевидно, что явное представление состояний важно. К счастью, сети Петри позволяют очень изящно представлять состояния в явном виде.

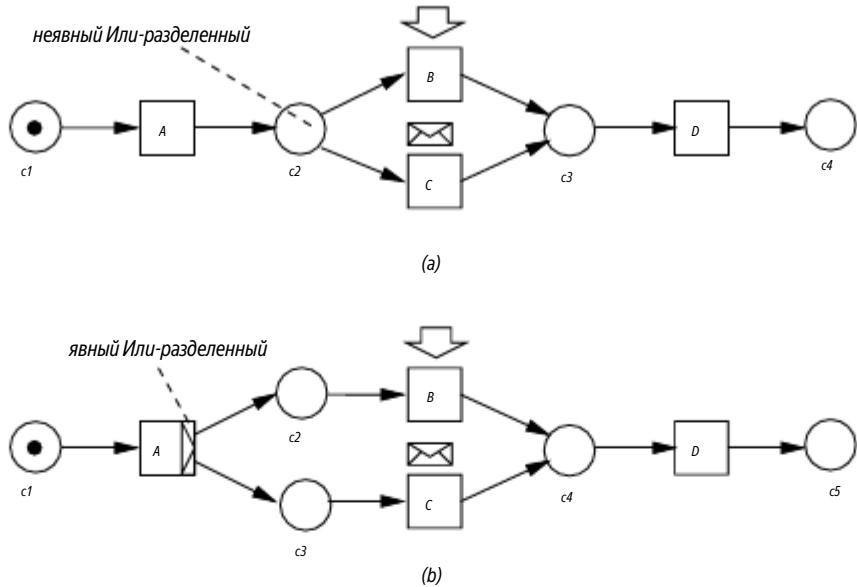


Рисунок 16: Пример, иллюстрирующий разницу между (а) неявным ИЛИ-разделением и (б) явным ИЛИ-разделением.

#### 4.4 Задачи, рабочие элементы и виды деятельности

Задача соответствует общей части работы. Задача определена не для конкретного случая, а для определенного типа случаев, т. Е. Задача может выполняться для многих случаев. Чтобы избежать путаницы между (1) задачей, (2) включением задачи и (3) выполнением задачи, мы вводим два дополнительных термина: *рабочий элемент* и *Активность*. Рабочий элемент - это задача, которая включена для конкретного случая, т.е. Рабочий элемент соответствует конкретному фрагменту работы. В некотором смысле рабочий элемент представляет собой комбинацию задачи, обращения и триггера (необязательно). В системе управления рабочими процессами рабочие элементы, соответствующие интерактивным задачам, предлагаются пользователю через так называемую корзину "Входящие". Действие - это фактическое выполнение рабочего элемента, т.е. Задача выполняется для конкретного случая. Действие сочетает в себе задачу, обращение, триггер и / или ресурс (необязательно). Легко сопоставить термины "задача", "рабочий элемент" и "действие" с терминологией сети Петри. Решение задач отвечает на переход, рабочий элемент соответствует разрешенному переходу (привязке) , а действие соответствует запуску перехода. Рассмотрим, например, рисунок 15. Переход process подачи жалобы соответствует задаче. Если оба  $c_5$  и  $c_7$  содержать to-ken с регистрационным идентификатором 91212123, затем разрешенный переход обработать жалобу соответствует рабочему элементу. Если обработать жалобу срабатывает при потреблении двух токенов с идентификатором регистра 91212123, тогда это срабатывание соответствует действию.

Можно думать о триггере как о дополнительном условии, которое должно быть выполнено прежде чем задача может быть выполнена. Если мы смоделируем это в терминах сетей Петри, триггер соответствует токену в дополнительном месте ввода задачи. Пока что у нас есть

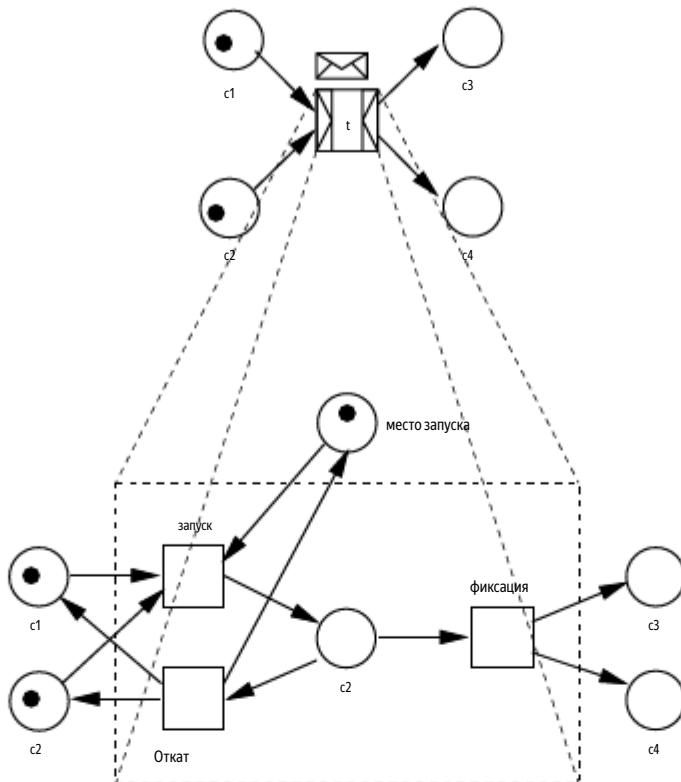


Рисунок 17: Низкоуровневое поведение задачи.

моделируемые задачи с помощью переходов. Сопоставляя задачи с переходами, мы абстрагируемся от внутреннего поведения задачи. Эта абстракция удобна для моделирования рабочего процесса . Однако, если мы рассмотрим задачу более подробно, то станет ясно, что выполнение задачи требует времени, т.е. между началом и завершением проходит время. На самом деле, возможно, что во время выполнения задачи происходят какие-то аномалии, например, происходит сбой приложения , база данных не работает или информация отсутствует. Следовательно, возможно, что выполнение задачи отложено или отменено. На рисунке 17 показана простая модель сети Петри низкоуровневого поведения задачи. За началом действия (переход *запуск* срабатывает) следует либо фиксация (переход *фиксация*), либо откат (переход *Откат*). Обратите внимание, что триггер моделируется маркером в месте *место запуска*.

Задачу можно рассматривать как *Логическую единицу работы* (LUW), а действие - как *trans-действие*. Как и транзакция, действие должно удовлетворять хорошо известным свойствам *ACID* . Аббревиатура *ACID* расшифровывается как:

*атомарность*

Действие выполнено успешно (*commit*) или полностью откатано, т.е. задача не может быть выполнена частично.

#### *согласованность*

Выполнение действия приводит к согласованному состоянию.

#### *изоляция*

Эффект от выполнения действия параллельно с другими видами деятельности равен эффекту от выполнения одного вида деятельности

#### *изолированно. долговечность*

Результат совершенного действия не может быть потерян.

Свойства ACID накладывают технические ограничения на размер задачи. Выбор задач "правильного" размера является одной из основных проблем при моделировании рабочих процессов. Если задачи слишком малы, то возникает много накладных расходов (время настройки) и рабочий поток становится слишком сложным для управления. Если задачи слишком велики, то может возникнуть проблема выполнить задачу за один раз, потому что невозможно зафиксировать частично выполненную задачу или передать части задачи на сокращение. Более того, управление ресурсами, т. е. Распределение рабочих элементов между пользователями, недостаточно детально проработано, если задачи слишком велики, чтобы допускать специализацию. Обратите внимание, что степень детализации управления ресурсами определяется размером каждой задачи.

## 4.5 Классификация ресурсов

Очевидно, что сети Петри хорошо подходят для моделирования определений рабочих процессов. Однако, до сих пор мы фокусировались на процессном аспекте, а не на аспекте ресурсов. Ресурсный аспект связан с организационными вопросами, такими как формирование команд и авторизация пользователей. Большинство систем управления рабочими процессами предоставляют инструмент проектирования для моделирования организации. Эти инструменты можно использовать для структурирования измерения ресурсов и облегчения спецификации правил, используемых для распределения рабочих элементов по ресурсам. Мы будем использовать термин *ресурс* для субъекта, который способен выполнять определенные задачи. В большинстве сред, где используются системы управления рабочими процессами, ресурсами являются в основном люди. Однако, как указано в разделе 2, мы предпочитаем термин "ресурс", а не "сотрудник" или "участник", потому что управление рабочими процессами не ограничивается офисами.

Для каждой задачи нам необходимо указать кому разрешено выполнять задачу. Однако, например, неявное связывание ресурсов и задач считается вредным. Если задача связана с конкретным пользователем, то задача будет заблокирована, если пользователь отсутствует. Использование ссылок на определенные ресурсы снижает гибкость. Кроме того, рабочий процесс дефиниции должен быть изменен каждый раз, когда новый сотрудник принят на работу (или существующий ЭМ-работника сбрасывается). Чтобы избежать ссылок на конкретные ресурсы, ресурсы сгруппированы по классам. А *класс ресурсов* - это группа ресурсов со сходными характеристиками.

Обратите внимание, что ресурс может быть членом нескольких классов ресурсов. Рассмотрим для примера рисунок 18. Существует четыре класса ресурсов: *Отдел жалоб*,

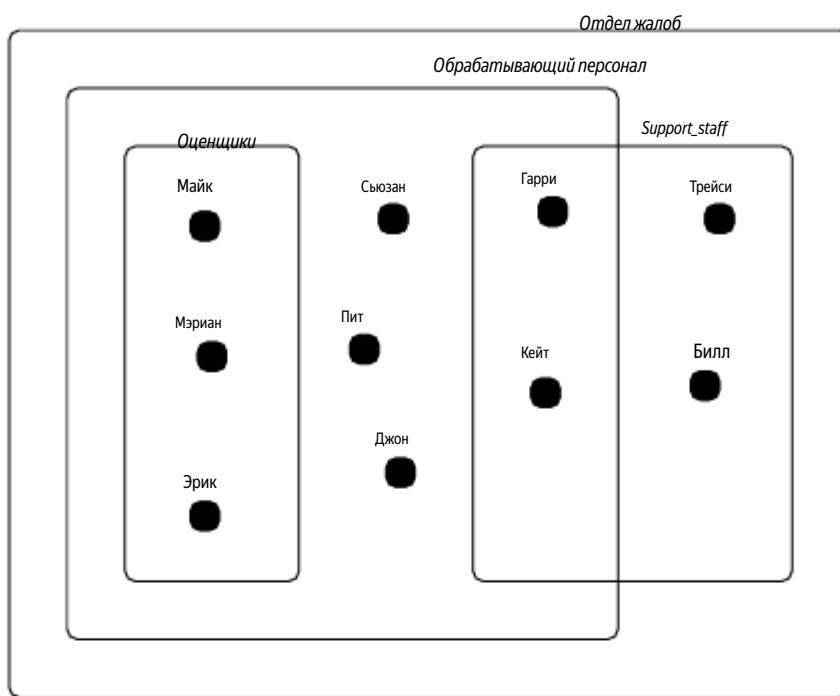


Рисунок 18: Классификация ресурсов.

*Оценщики, Обслуживающий персонал и Вспомогательный персонал. Мэриан является членом Оценщики, Обслуживающий персонал и Отдел рассмотрения жалоб, но она не является членом класса re-source Вспомогательный персонал. Классификация ресурсов, показанная на рисунке 18, может быть использована для указания того, какие ресурсы разрешены для выполнения четырех пользовательских задач, показанных на рисунке 15. Задачи Зарегистрироваться, оценить, обработать жалобу и обработка чеков должна – выполняться членом класса Отдел жалоб. Кроме того,, для регистрации – ter требуется член класса Вспомогательный персонал, оценивает требуется член класса Оценщики, обрабатывают жалобу требуется член класса Обрабатывающий персонал и обработка чеков требуется член класса Оценщики. Обратите внимание, что без явной привязки задач и ресурсов указано, что Мэриан является одним из трех человек, которым разрешено выполнять задачу проверка обработки.*

*В разделе 2, два вида ресурсов были введены классы: роли и организационные единицы. Если класс ресурсов основан на возможностях (т.е. функциональных требованиях) его членов, он называется Роль. Классы ресурсов Вспомогательный персонал, Оценщики и Обрабатывающий персонал являются примерами ролей. Если классификация основана на структуре организации, такой класс ресурсов называется организационным подразделением (например, командой, филиалом или отделом). Класс Отдел рассмотрения жалоб является примером класса ресурсов, который основан на структуре организации, а не на возможностях ее членов. Многие системы управления рабочими процессами связывают роль и организационное подразделение с каждой задачей. Это означает, что задача должна быть*

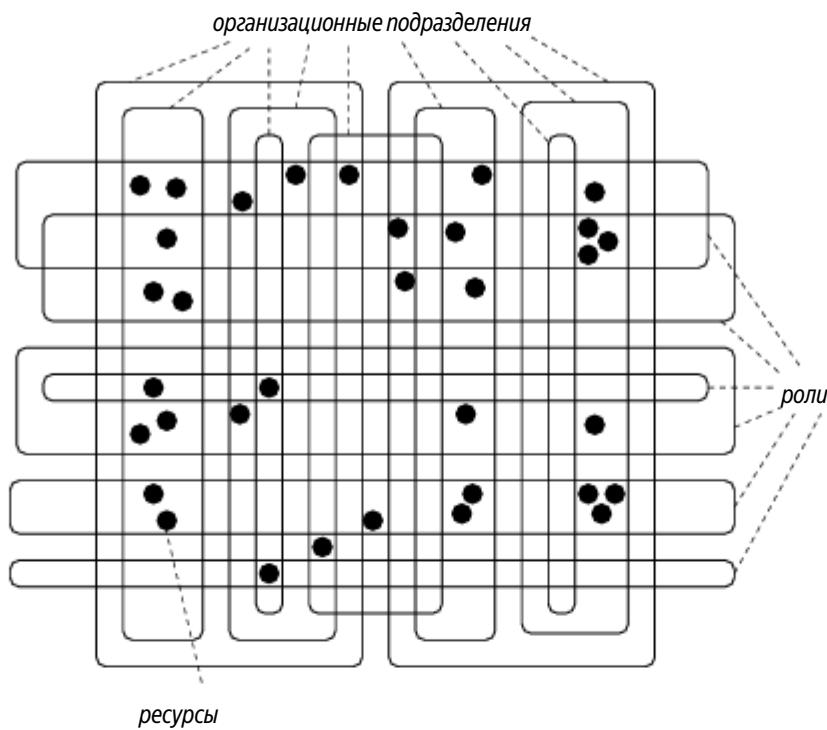


Рисунок 19: Классификация ресурсов: ресурс является членом организационных подразделений и ролей.

выполняется ресурсом, который является членом как класса ресурсов, основанного на функциональных требованиях (ролях), так и класса ресурсов, основанного на структуре организации (организационной единицы). Рисунок 19 иллюстрирует ортогональную связь между ролями и организационными подразделениями.

Связывая роль и организационную единицу с каждой задачей, количество ресурсов, разрешенных для выполнения этой задачи, сокращается до пересечения обоих классов. Иногда требуются более сложные концепции, чтобы указать, кому разрешено выполнять конкретный рабочий элемент. Иногда роль или организационное подразделение, связанное с задачей, зависит от регистра. Например, страховые претензии с предполагаемой стоимостью компенсации более 1000 долларов рассматриваются высокопоставленным сотрудником, или жалобы на конкретный отдел рассматриваются командой А. Также возможно, что определенные ресурсы исключены, потому что они выполнили связанные задачи. Например, задачи *оценивают и проверяют обработку* на рисунке 15 должны выполняться разными ресурсами. Это означает, что если Мэриан выполнила задачу *оценить* для конкретного случая, ей не разрешено выполнять *проверка обработки* для того же случая (т.е. Это должно выполняться Майком или Эриком). Мы используем термин "разделение функций" для этого дополнительного требования.

## 4.6 Управление ресурсами

Классификация ресурсов и более продвинутые концепции, такие как разделение функций, используются для указания того, кому разрешено выполнять рабочий элемент. Если для выполнения рабочего элемента разрешено несколько ресурсов, необходимо сделать выбор. Для решения этого вопроса есть два механизма:

### *нажимное управление*

Система управления рабочим процессом делает выбор и отправляет каждый рабочий элемент конкретному пользователю, т.е. для каждого рабочего элемента

выбирается ресурс. Выбор может основываться на статистической информации, рабочей нагрузке или простой эвристике. *управление перемещением*

Система управления рабочими процессами отправляет копию каждого рабочего элемента каждому пользователю, которому разрешено его выполнять. В тот момент, когда один пользователь выбирает рабочий элемент, все остальные копии исчезают.

В большинстве ситуаций pull control является предпочтительным механизмом распределения рабочих элементов. Обратите внимание, что push control снижает гибкость, поскольку привязывает рабочие элементы к определенным ресурсам. Пользователям системы управления рабочими процессами часто не нравится принудительное управление потому что они чувствуют, что система контролирует их. Выбор между несколькими ресурсами, разрешенными для выполнения рабочего элемента, не является единственным решением, которое необходимо принять. Если имеется много рабочих элементов, необходимо определить порядок, в котором эти рабочие элементы должны быть выполнены. Для упорядочивания ожидающих выполнения рабочих элементов могут использоваться хорошо известные схемы организации очередей, известные из operations management ([Wil89]): FIFO (поступление первым), LIFO (отправление последним), SPT (наименьшее время обработки), SRPT (наименьшее оставшееся время обработки), EDD (самый ранний срок выполнения) и PRIO (задачи с приоритетом идут первыми). Если используется принудительное управление, система управления рабочим процессом будет принудительно выполнять рабочие элементы в определенном порядке. При использовании pull-управления пользователь может отклониться от порядка, предложенного системой. Обратите внимание, что порядок расположения рабочих элементов и выбор ресурсов тесно связаны.

Каждый раз, когда пользователь выполняет рабочий элемент, требуется настройка как для конкретного случая, так и для задачи. Для сокращения времени настройки некоторые системы управления рабочими процессами допускают *последовательное выполнение* и *поэтапное выполнение*. Последовательное выполнение соответствует выполнению множества последующих задач для одного и того же случая за один раз и одним пользователем. Рассмотрим для примера бизнес-процесса показана на рис.15. Если задачи *Регистрация*, *отправить анкету* и *оценить* может быть выполнено определенным ресурсом, тогда цепное выполнение означало бы, что если ресурс запускает задачу *Зарегистрироваться*, две другие задачи будут выполняться автоматически (для того же случая). Последовательное выполнение позволяет группировать задачи в макрозадачи. Групповое выполнение соответствует повторному согласованному выполнению одной задачи для нескольких случаев. Например, ресурс выполняет

задача *Зарегистрироваться* для пакета обращений. Последовательное выполнение сокращает общую настройку времени, необходимое для ознакомления с обращением. Поэтапное выполнение помогает пользователю создавать рутину. Концепция, которая тесно связана с последовательным выполнением, - это *управление делами*. Если используется управление обращениями, то у каждого обращения есть менеджер обращений. Менеджер обращений отвечает за обращение и выполняет как можно больше задач для обращения. Использование системы управления обращениями имеет два преимущества: сокращается время настройки и менеджер по обращениям служит контактным лицом для клиента, чье дело рассматривается. Лишь немногие системы управления рабочими процессами поддерживают передовые концепции, такие как последовательное выполнение, поэтапное выполнение и управление обращениями.

#### 4.7 Краткие сведения

В этом разделе мы представили ряд концепций рабочего процесса и показали, как эти концепции могут быть отображены в сетях Петри. Четыре конструкции маршрутизации, идентифицированные WfMC, были отображены в WF-сетях. Формализм сети Петри также позволил нам ощутить тонкую разницу между неявным ИЛИ-расщеплением и явным ИЛИ-расщеплением. Более сложные темы, такие как контрольные точки, триггеры и разница между задачами, рабочими элементами и действиями, были проиллюстрированы с помощью нескольких примеров. Эти примеры показывают, что формализм сети Петри хорошо подходит для работы с первыми двумя измерениями, показанными на рисунке 2. Установить связь между определением рабочего процесса, определенным в терминах WF-сети, и третьим измерением (измерение ресурсов), мы обсудили классификацию ресурсов и распределение ресурсов по рабочим элементам (*управление ресурсами*).

### 5 Анализ рабочих процессов

В этом разделе мы сосредоточимся на анализе рабочих процессов. Мы начнем с обзора различных типов анализа рабочего процесса. Затем мы сосредоточимся на конкретном типе анализа: проверке. Для этой цели мы введем понятие корректности, называемое "обоснованностью", и посвятим оставшуюся часть этого раздела анализу на основе сети Петри, методам установления обоснованности. Результаты, представленные в этом разделе, проиллюстрируют потенциал сетей Петри в области рабочих процессов.

#### 5.1 Введение

Корректность, эффективность и действенность бизнес-процессов, поддерживаемых системой управления рабочими процессами, жизненно важны для организации. Рабочий процесс определение процесса, содержащее ошибки, может привести к недовольству клиентов, повторному ведению журнала, искам о возмещении ущерба и потере репутации. Недостатки в разработке определения рабочего процесса

это также может привести к увеличению пропускной способности, снижению уровня обслуживания и необходимости в избыточной емкости. Вот почему важно проанализировать определение рабочего процесса должно быть до его запуска в производство. В принципе, существует три типа анализа:

*валидация*, то есть проверка того, работает ли рабочий процесс так, как ожидалось,

*проверка*, т.е. установление правильности рабочего процесса, и

*анализ производительности*, т.е. оценка способности соответствовать требованиям с учетом времени пропускной способности, уровней обслуживания и использования ресурсов.

Проверка может быть произведена с помощью интерактивного моделирования: несколько фиктивных обращений загружаются в систему, чтобы проверить, хорошо ли они обрабатываются. Для верификации и анализа эффективности управления необходимы более продвинутые методы анализа. К счастью, для сетей Петри было разработано много мощных методов анализа ([Mur89]). Линейные алгебраические методы могут быть использованы для проверки многих свойств, например, инвариантов места, инвариантов перехода и (не) достижимости. Анализ графика покрытия, проверка модели и методы сокращения могут быть использованы для анализа динамического поведения сети Петри. Моделирование и анализ марковской цепи могут быть использованы для оценки производительности (см. [MBea95]). Обилие доступных методов анализа показывает, что сети Петри можно рассматривать как независимое от решателя средство между разработкой определения рабочего процесса и анализом рабочего процесса. Современные системы управления рабочими процессами предоставляют ограниченную поддержку анализу производительности. Большинство систем управления рабочими процессами предоставляют элементарный симулятор или обеспечивают доступ к инструменту моделирования. Моделирование может использоваться для оценки ключевых показателей эффективности: среднее время обработки обращений, среднее время ожидания, уровень использования ресурсов, уровни обслуживания и среднее количество незавершенных обращений. Большинство систем управления рабочими процессами не предоставляют никакой поддержки для проверки рабочих процессов. В результате, определения процессов workflow вступают в силу до того, как они будут тщательно проверены на правильность. Это часто приводит к ошибкам во время выполнения, которые необходимо исправлять "на лету" с большими затратами. Как производители, так и пользователи систем управления рабочими процессами видят потребность в инструментах анализа, которые заботятся о проверке рабочих процессов. К сожалению, большинство производителей не располагают технологиями для создания таких инструментов. В оставшейся части этого раздела мы покажем, что рабочие процессы, заданные в терминах сетей Петри, могут быть проверены с использованием самых современных методов анализа.

## 5.2 Свойство надежности

Рассмотрим WF-сеть, показанную на рисунке 20. Легко видеть, что это определение процесса рабочего процесса содержит несколько недостатков. Если время ожидания 1 и обработка 2 срабатывают, или тайм-аут 2 и обработка 1 срабатывают, тогда WF-сеть не завершится должным образом потому что токен застревает в  $c_5$  или  $c_4$ . Если время ожидания 1 и время ожидания 2 сработают, то задача обработка NOK будет выполнена дважды и из-за наличия двух токенов в момент завершения не ясен.

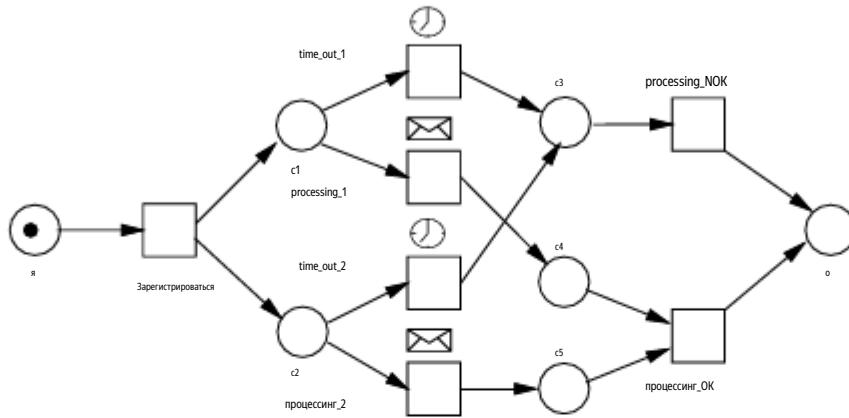


Рисунок 20: Еще одна WF-сеть для обработки жалоб.

Очевидно, что WF-сеть, показанная на рисунке 20, некорректна, но каково определение корректности? Критерий корректности, который мы используем в этой статье, представляет собой набор минимальных требований, которым должно удовлетворять любое определение рабочего процесса. Прежде всего, определение процесса work-flow должно удовлетворять двум требованиям, перечисленным в Определении 6:

- (1) WF-сеть имеет место источника (начальное условие) и место приемника (конечное условие),
  - (2) каждая задача / условие находится на пути от к. Эти требования могут проверяться статически, т.е. они относятся только к структуре сети Петри.
- Однако, есть третье требование, которое должно быть выполнено:

*В любом случае, процедура рано или поздно завершится, и в том момент, когда процедура завершается, на месте о стоит токен, а все остальные места пусты.*

Более того, существует четвертое требование:

*Не должно быть мертвых задач, т. е. Должна быть возможность выполнить произвольную задачу, следуя соответствующему маршруту через WF-сеть.*

Эти два дополнительных ограничения соответствуют так называемому свойству обоснованности.

**Определение 7 (Звук)** Процедура,  
моделируемая WF-сетью тогда и только тогда, когда:

PN  $\vdash_{TF}$  является правильной

(i) Для каждого состояния  $M$  достижимого из состояния, существует опережение последовательности запуска-<sup>2</sup> поступающий от государства  $M_0$  в государству. Формально:

$$8_{M^c} \text{ я ! } M_i ) M_i ! o_j$$

(ii) Состояние - это единственное состояние, доступное из состояния, в котором установлен хотя бы один токен  $o$ . Формально:

$$8_{M^c} \text{ я ! } M^c \wedge M ) o = o_i M_i$$

(iii) В системе отсутствуют мертвые переходы ( PN  $\vdash_{TF}$ ). Формально:

$$8_{T_T} 9_{MM^0} \text{ я ! } M !_t M^0$$

Обратите внимание, что свойство надежности относится к динамике WF-сети. Первое требование в Определении 7 гласит, что, начиная с начального состояния (state  $\text{я}$ ), всегда возможно достичь состояния с одним токеном на месте  $o$  (состояние  $o$ ). Если мы как- предполагаем справедливость (ср. [Val87, Mur89]), то первое требование подразумевает, что в конечном итоге- союзное государство будет достигнуто. Предположение о справедливости разумно в контексте управления рабочими процессами: все решения принимаются (неявно или явно) приложениями , людьми или внешними субъектами. Очевидно, что они не должны вводить бесконечный цикл. (Мы вернемся к вопросу справедливости.) Второе требование в Определении статьи 7 гласит, что в момент установки токена  $o$ , все остальные места должны быть пустыми. Иногда термин *надлежащее завершение* используется для описания первых двух требований [GCEV72]. Третье и последнее требование гласит, что нет

мертвые переходы (задачи) в исходном состоянии  $\text{я}$ , т.е. для каждого перехода возможно чтобы достичь (начиная с) состояния, в котором включено.

Определение 7 предполагает, что рабочий процесс задан в терминах WF-сети. Напомним, что WF-сеть - это классическая сеть Петри, которая удовлетворяет определенным свойствам. Тем не менее, мы расширили определение рабочего процесса с помощью триггеров и атрибутов рабочего процесса . Методы анализа, представленные в оставшейся части этой статьи, будут абстрагироваться от этой дополнительной информации. Триггеры просто удаляются. И- разделение/И-объединение соответствует обычному переходу в сети Петри. Маркеры не имеют значения (неокрашены). Однако мы не можем сопоставить задачу "ИЛИ-разделить" и / или "ИЛИ-объединить" с одним переходом. ИЛИ-разделение развертывается в небольшую сеть с переходом для каждой альтернативы. ИЛИ-объединение также заменяется двумя или более переходами.

---

<sup>2</sup> Обратите внимание, что имеет место перегрузка обозначения: символ  $\text{я}$  используется для обозначения как *места*  $i$ , так и *состояния* только с одним маркером на месте  $\text{я}$  (см. Раздел 3).

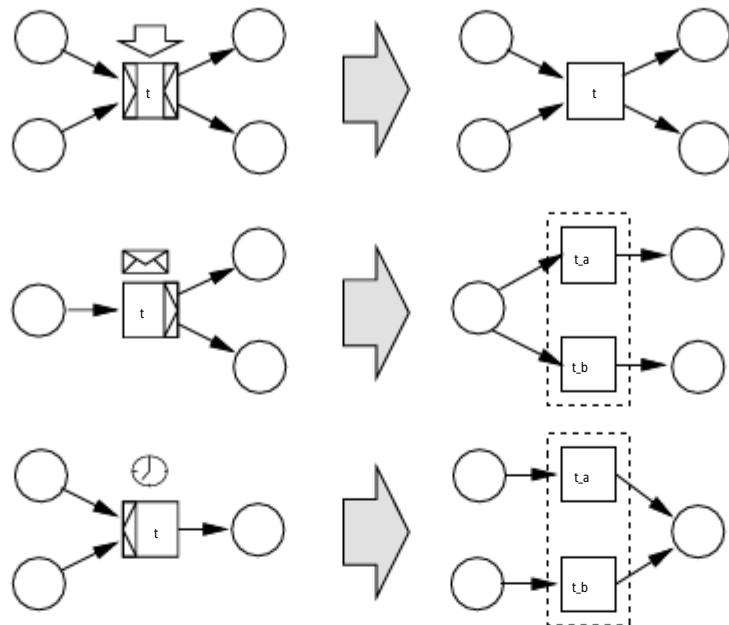


Рисунок 21: Удаление триггеров и развертывание задач для получения WF-сети.

На рисунке 21 показан перевод в WF-сеть путем удаления триггеров и развертывания задач. Существует несколько причин для абстрагирования от триггеров и атрибутов рабочего процесса. Прежде всего, поведение среды невозможно полностью смоделировать. Среда (например, сотрудники и клиенты) генерирует триггеры и устанавливает атрибуты рабочего процесса. Для целей моделирования необходимо в определенной степени смоделировать поведение окружающей среды.

Однако для целей проверки очень опасно использовать явно смоделированную среду. Если реальная среда отличается от смоделированной среды, проверка бессмысленна. Неправильные предположения о среде могут скрывать ошибки при разработке определения рабочего процесса. Поэтому мы абстрагируемся от триггеров и атрибутов рабочего процесса. Мы просто предполагаем, что среда готова генерировать триггеры, и каждый выбор считается недетерминированным. Рассмотрим, например, задачу *оценить* на рисунке 15. Эта задача заменена двумя переходами для моделирования двух возможных результатов оценки. В действительности выбор основан на атрибутах рабочего процесса, данных приложения и /или поведении оценщика. Очевидно, что это невозможно полностью смоделировать. Поэтому поэтому мы считаем этот выбор недетерминированным. Существуют и другие причины для абстрагирования от триггеров и атрибутов рабочего процесса. Если мы сможем доказать обоснованность ситуации без триггеров и атрибутов рабочего процесса, это также будет справедливо для ситуации с триггерами и атрибутами рабочего процесса. И последнее, но не менее важное: мы абстрагируемся от триггеров и атрибутов рабочего процесса, потому что это позволяет нам использовать классические сети Петри вместо высокогруженных сетей Петри. Анализ точки зрения, расширения Clas-sical сеть Петри является предпочтительным ввиду наличия эффективных алгоритмов и

мощные инструменты анализа.

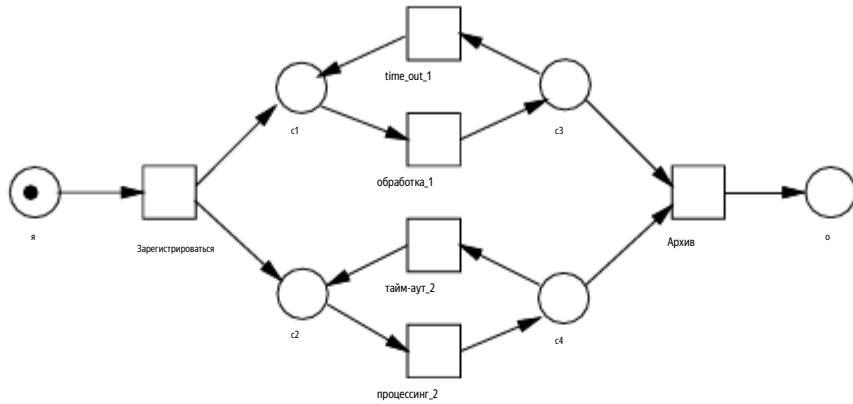


Рисунок 22: Надежная WF-сеть, иллюстрирующая предположение о справедливости.

Определение 7 устанавливает требования, гарантирующие возможность прекращения,

т. е., глядя в состояние  $s$  всегда возможно достичь конечного состояния. Рассмотрим для

$s$

пример WF-сети, показанной на рисунке 22. Из любого состояния, доступного из состояния  $s$ ,

$s$

существует последовательность запуска, которая приводит к состоянию  $s$ . Однако это не гарантирует, что последовательность срабатывания будет выполнена. Например, возможно, что всегда один из тайм-аутов наступает перед выполнением задачи. *Архив* происходит. Следовательно, мы предполагаем *сильную справедливость* (ср. [Val87, Mur89]). Это означает, что если мы рассмотрим произвольную бесконечную последовательность запуска расширенной сети (см. Рисунок 23), то каждый переход происходит в-

конечно, часто. (В расширенной сети добавлен дополнительный переход  $t$ , который соединяет

$s$

и, таким образом, делая сеть циклической.) Сильный справедливости соответствует probabilistic толкование справедливости, например, включение времени выполнения каждой задачи негативных экспоненциально распределенных, или включены задачи подобраны в случайном порядке (ср. МФ. [MBea95]).

Предполагая обоснованность и строгую справедливость, гарантируется, что каждое дело завершится

в конце концов, все  $s$  станут на свои места. Более слабой формы справедливости часто бывает

недостаточно. Рассмотрим, например, следующее понятие справедливости: переход, который включается бесконечно часто, в конечном итоге сработает. Если мы примем это понятие справедливости, оно не гарантирует-

указано, что это состояние  $s$  будет достигнуто на рисунке 22. Если задачи выполнены *обработка 1, время ожидания 1, –*

$- - -$

*обработка 2, и тайм-аут 2* запускайте позже, затем выполняйте задание *Архив* никогда не станет

включенным, и, несмотря на предположение о слабой справедливости, завершение не гарантировано

. Однако разумно исходить из строгой справедливости в контексте управления рабочим процессом

. Более того, довольно легко обнаружить поведение, которое нарушает

принцип строгой справедливости, например, подсчитать, сколько раз выполняется каждая задача

для конкретного случая.

### 5.3 Необходимое и достаточное условие надежности

Учитывая WF-сеть  $\overline{PN} = \overline{P} \cup \overline{T}$ , мы хотим решить, является ли  $\overline{PN}$  исправным. Для этой цели мы определяем расширенную сеть  $\overline{P} \cup \overline{T} \cup \{f_t\}$ , где  $\overline{PN}$  — сеть Петри и получается путем добавления дополнительного перехода  $f_t$ , который соединяет  $P$  и  $T$ . Расширенный сеть Петри  $\overline{PN} = \overline{P} \cup \overline{T} \cup \{f_t\}$  определяется следующим образом:  $P = P_T, T = T \cup f_t, f_t \in \overline{F}$ . Рисунок 23 иллюстрирует связь между  $\overline{PN}$  и  $\overline{P}$ .  $\overline{PN}$  может использоваться для облегчения проверки надежности расширенной собственности. Следующая теорема показывает, что расширенная сеть позволяет формировать свойство надежности в терминах хорошо известных свойств сети Петри.

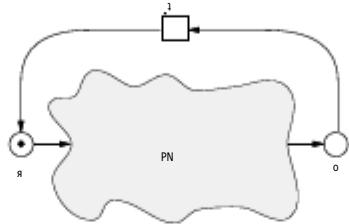


Рисунок 23:  $\overline{PN} = \overline{P} \cup \overline{T} \cup \{f_t\}$  и  $\overline{f_t} \in \overline{F}$  и  $\overline{\phi}_{x_0 t}$  и  $\chi_{T_f}$ .

**Теорема 1** WF-сеть

$\overline{PN}$  является звуковым тогда и только тогда, когда  $\overline{PN}$  является живым и ограниченным.

**Доказательство.**

Смотрите [Aal97] или [Aal96a].

Эта теорема показывает, что стандартные методы анализа на основе сети Петри могут быть использованы для проверки достоверности. Рассмотрим, например, определение рабочего процесса, показанное на рисунке 15. Мы можем использовать один из многих стандартных инструментов анализа на основе сетей Петри, чтобы убедиться, что расширенная сеть является активной и ограниченной. Следовательно, рабочий процесс, указанный на рисунке 15, гарантированно работает должным образом (см. Определение 7). Теорема 1 является продолжением результатов, представленных в [Aal95, SH95]. В [Aal95] мы повторно ограничиваем себя свободными сетями выбора. В сетях Петри со свободным выбором не допускается смешивать выбор и синхронизацию (см. Определение 8). Независимо друг от друга Штрауб и Хуртадо [SH95] нашли необходимые и достаточные условия для надежности КОПА сетей. (Сети COPA соответствуют подклассу сетей Петри со свободным выбором.)

### 5.4 Структурные характеристики надежности

Теорема 1 дает полезную характеристику качества определения рабочего процесса. Однако существует ряд проблем.:

Для сложной WF-сети может оказаться непростым решить, насколько она надежна.

Для Arbi-trary WF-сетки обоснованность разрешим, но и дорого с точки зрения времени и пространства сложности. На самом деле, проблема определения живучести и ограниченности сложна с точки зрения EXPSPACE (см. [CEP93]). Надежность - это минимальное требование. Проблемы удобочитаемости и ремонтопригодности не рассматриваются теоремой 1. Теорема 1 не показывает, как следует модифицировать неработоспособную WF-сеть, т.е. Она не идентифицирует конструкции, которые делают недействительным свойство надежности.

Эти проблемы происходят из того факта, что определение надежности относится к динамике WF-сети, в то время как разработчик рабочего процесса имеет дело со статической структурой WF-сети. Поэтому интересно исследовать структурные характеристики-terizations звука WF-сеткой. Для этой цели мы вводим три интересных подкласса WF-сетей: WF-сети со свободным выбором, хорошо структурированные WF-сети и S-хватываемые WF-сети.

#### 5.4.1 WF-сети со свободным выбором

Большинство систем управления рабочими процессами, доступных на данный момент, абстрагируются от состояний между задачами, т.е. Состояния не представлены явно. Эти системы управления рабочим процессом используют стандартные блоки, такие как И-разделить, И-объединить, ИЛИ-разделить и ИЛИ-объединить, чтобы указать процедуры рабочего процесса. Для параллельной маршрутизации используются команды AND-split и AND-join. Для условной маршрутизации используются команды OR-split и OR-join. Поскольку эти системы абстрагируются от состояний, каждый выбор делается внутри строительного блока "ИЛИ-разделить". Если мы моделируем разделение OR в терминах сети Петри, разделение OR соответствует ряду переходов, использующих один и тот же набор входных данных (см. Рис. 21). Это означает, что для этих систем управления рабочими процессами, процедура рабочего процесса соответствует а *сеть Петри со свободным выбором*.

**Определение 8 (Свободный выбор)** Сеть Петри - это сеть Петри со свободным выбором, если для каждого двух переходов  $1^t$  и  $2^t$  имеется подразумеваемый переход  $1^t \parallel 2^t$

Легко видеть, что определение процесса, состоящее из AND-splits И-joins, ИЛИ-разделяется и ИЛИ-присоединяется - это свободный выбор ([Aal97]). Если два перехода 1 и 2 поделитесь местом ввода ( $1^t \parallel 2^t$ ), то они являются частью разделения ИЛИ, то есть "свободного выбора" между рядом альтернатив. Следовательно, наборы входных мест для 1 и 2 должны соответствовать ( $1^t \parallel 2^t$ ). На рисунке 20 показана WF-сеть со свободным выбором. WF-сеть = показанное на рисунке 15 не является свободным выбором: Архив и обработать жалобу используется совместно с местом ввода, но два соответствующих набора входных данных различаются.

Мы оценили многие системы управления рабочими процессами, и лишь некоторые из них

системы (например, COSA, INCOME и LEU) допускают конструкцию, которая сравнима с WF-сетью несвободного выбора. Следовательно, имеет смысл рассмотреть сети Петри со свободным выбором. Очевидно, что параллелизм, последовательная маршрутизация, условная маршрутизация и итерация могут быть смоделированы без нарушения свойства свободного выбора. Другой причиной ограничения WF-сетей сетями Петри со свободным выбором является следующее. Если мы допустим сети Петри со свободным выбором, то выбор между конфликтующими задачами *может* на вас будет влиять порядок выполнения предыдущих задач..... Маршрутизация обращения должна быть независимой от порядка, в котором выполняются задачи. Ситуация, в которой нарушаются свойство свободного выбора, часто представляет собой смесь параллелизма и выбора.

На рисунке 24 показана такая ситуация. Запускающий переход  $t_1$  вводит параллелизм. Альтернативный выбор между  $t_2$  и  $t_5$  не введен ( $t_5$  не включен), параллельное выполнение  $t_2$  и  $t_3$  приводит к ситуации, когда  $t_5$  не допускается. Каким образом когда-либо, если выполнение  $t_2$  откладывается до  $t_3$  был выполнен, тогда есть реальный выбор между  $t_2$  и  $t_5$ . По нашему мнению, сам параллелизм следует отделять от выбора между двумя или более альтернативами. Следовательно, мы считаем конструкцию несвободного выбора, показанную на рисунке 24, неправильной. В литературе термин *путаница* часто используется для обозначения ситуации, показанной на рисунке 24.

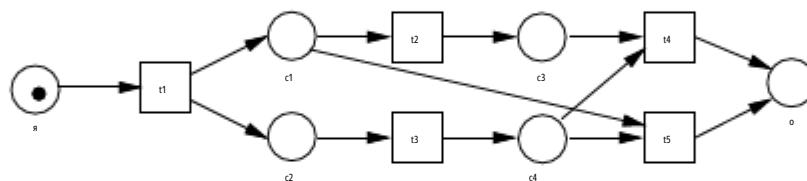


Рисунок 24: WF-сеть без свободного выбора, содержащая смесь параллелизма и выбора.

Сети Петри со свободным выбором были широко изучены (см. [Bes87, DE95, Des92, Esp90, Hac72]), потому что они кажутся хорошим компромиссом между выразительностью мощности и анализируемости. Это класс сетей Петри, для которых существуют сильные теоретические результаты и эффективные методы анализа. Например, хорошо известная теорема о ранге ([DE95]) позволяет нам сформулировать следующее следствие.

**Следствие 1** WF-сеть со свободным выбором может быть проверена на надежность за полиномиальное время.

**Доказательство.**

Смотрите [Aal96a].

Следствие 1 показывает, что для сетей со свободным выбором существуют эффективные алгоритмы для определения обоснованности. Также можно показать, что надежные WF-сети со свободным выбором безопасны ([Aal96a]).

Хотя большинство систем управления рабочими процессами допускают только рабочие потоки со свободным выбором, WF-сети со свободным выбором не являются полностью удовлетворительной структурной характеристикой "хороших" рабочих процессов. .... С одной стороны, существуют WF-сети со свободным выбором которые соответствуют разумным рабочим процессам (см. Рисунок 15). С другой стороны, существуют надежные WF-сети со свободным выбором, которые не имеют смысла. Тем не менее, свойство свободного выбора является желательным свойством. Если можно смоделировать рабочий процесс как WF-сеть со свободным выбором, то следует это сделать. Спецификация рабочего процесса, основанная на WF-сети свободного выбора, может быть введена в действие большинством систем рабочего процесса. Более того, WF-сеть со свободным выбором позволяет использовать эффективные методы анализа и проста для понимания. Конструкции без свободного выбора, такие как конструкция, показанная на рис. 24, являются потенциальным источником аномального поведения (например, взаимоблокировки), которое трудно отследить.

#### 5.4.2 Хорошо структурированные WF-сети

Другой подход к получению структурной характеристики "хороших" рабочих процессов заключается в балансировании и / Или разделении и И / Или объединении. Очевидно, что два параллельных потока, инициированных разделением И, не должны соединяться соединением ИЛИ. Два альтернативных потока, созданных с помощью OR-split, не должны синхронизироваться с помощью AND-join. Как показано на рисунке 25, разделитель "И" должен быть дополнен соединением "ИЛИ", а разделитель "ИЛИ" должен быть дополнен соединением "ИЛИ".

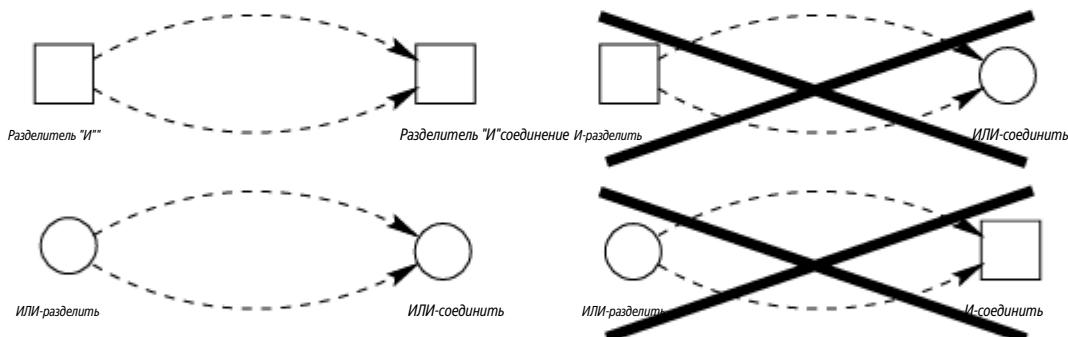


Рисунок 25: Хорошие и плохие конструкции.

Одним из недостатков WF-сети, показанной на рисунке 20, является тот факт, что И-разделяется. Зарегистрироваться дополняется символом ИЛИ-join с3 или символом ИЛИ-join o. Чтобы формализовать концепцию, проиллюстрированную на рисунке 25, мы даем следующее определение.

**Определение 9 (Хорошо управляемый, хорошо структурированный) Сеть Петри**

для любой пары узлов  $x \text{ и } y$  такой, что один из узлов является местом, а другой

а переход и для любой пары элементарных путей

$$\forall g \in \text{ПН} \exists x, y \in \text{С} \quad x \text{ и } y \text{ - элементарные пути} \quad \text{если для}$$

расширенная сеть

хорошо обрабатывается, если:

$\text{PN}$

$x \text{ и } y \text{ для, 2 ведущий из}$

$\text{C}_1 \text{ и } \text{C}_2 \text{ для}$

$\text{PN}$  хорошо структурирован, если для

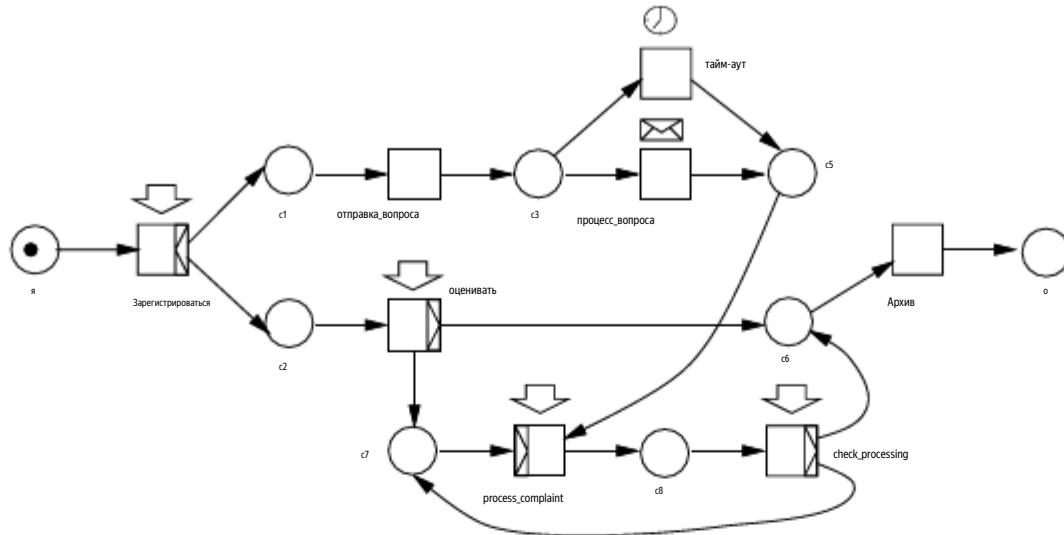


Рисунок 26: Определение рабочего процесса, которое плохо структурировано.

Рассмотрим, например, альтернативный процесс рассмотрения жалоб, показанный на рис.

уре 26. WF-net не очень хорошо структурирован, поскольку И-разделен *Зарегистрироваться* дополняется соединением OR- сб: есть два непересекающихся пути, ведущих из *Зарегистрироваться*. Для сб, один через с2 и оцените, и один через с1, отправьте анкету, с3, обработайте задание- участник, с5, обработайте жалобу, с8 и проверьте обработку. Эти два пути подчеркивают тот факт, что рабочий процесс, показанный на рисунке 26, не является надежным. Если задача оценка помещает маркер в сб, тогда токен попадает в ловушку на втором пути (например, в месте с5). Если дуга из с5 Для жалоба на процесс заменяется дугой из с5.

Для *Архив*, затем разделите И *Зарегистрироваться* дополняется символом AND-присоединиться *Архив* и WF-сеть исправна.

На рисунке 20 показан другой пример плохо структурированной WF-сети. Сеть WF- плохо структурирована, потому что И-split *Зарегистрироваться* дополняется символом с3.

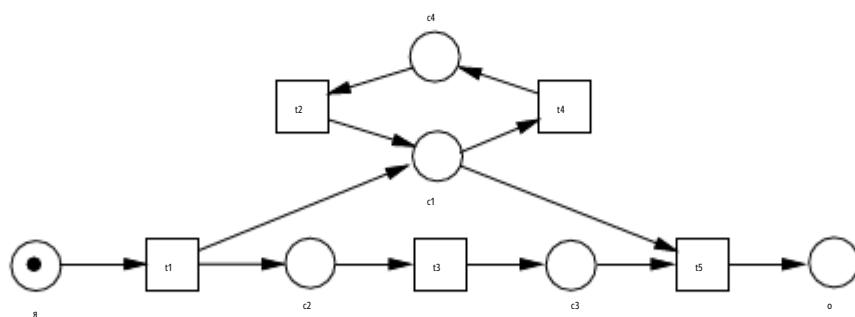


Рисунок 27: Хорошо структурированная WF-сеть.

На рисунке 27 показан типичный пример хорошо структурированной WF-сети, которая не является свободной по выбору. WF-сеть, показанная на рисунке 22, является еще одним примером хорошо структурированной сети

что не является свободным выбором. Хотя обе WF-сети не являются свободными, можно очень эффективно проверить надежность такой WF-сети.

**Следствие 2** Хорошо структурированный WF-продажи могут быть проверены на устойчивость во полу-Миал времени.

**Доказательство.**

См. [Aal96a]. В доказательстве используются результаты, представленные в [ES90] и [BCD95].

Хорошо структурированные WF-сети и WF-сети со свободным выбором обладают схожими свойствами. В обоих случаях надежность может быть проверена очень эффективно, а надежность подразумевает безопасность. Несмотря на эти сходства, существуют надежные, хорошо структурированные WF-сети, которые не имеют свободного выбора (рисунок 27), и существуют надежные, не хорошо структурированные WF-сети со свободным выбором. На самом деле, возможно иметь надежную WF-сеть, которая не является ни свободной в выборе, ни хорошо структурированной (рисунки 15 и 24).

#### 5.4.3 S-покрываемые WF-сети

Как насчет звуковых WF-сетей, показанных на рис. 15 и 24? WF-сеть, показанную на рисунке 24, можно преобразовать в хорошо структурированную WF-сеть со свободным выбором, разделив выбор и параллелизм. WF-сеть, показанная на рисунке 15, не может быть преобразована в WF-сеть со свободным выбором или хорошо структурированной структурой без получения гораздо более сложной WF-сети. Разместить  $c_5$  действует как своего рода веха, которая проверяется задачей обработать жалобу. Традиционные системы управления рабочими процессами, которые не делают состояния обращения явным, не способны обрабатывать рабочий процесс, указанный на рисунке 15. Только системы управления рабочими процессами, такие как COSA ([SL96]), имеют возможность реализовать такой рабочий процесс на основе состояния. Тем не менее, интересно рассмотреть обобщения WF-сетей со свободным выбором и хорошей структурой:

S-охватываемые WF-сети можно рассматривать как такое обобщение.

**Определение 10 (S-покрываемая) WF-сеть**

$\text{PN}_s$  является S-покрываемой, если расширенная сеть

$\text{P}_s \text{T}_s \Phi_s$  удовлетворяет следующему свойству. Для каждого места существует подсеть  $\text{PN}_{s'}$ ,  $\text{P}_s \text{PN}_{s'}$

$\text{P}_s \text{T}_s \text{F}_s$  такой, что:  $\text{p} \in \text{Z}_{\prod_{s'} \text{P}_s \text{PN}_{s'}}$   $\text{PT}_s$ ,  $\text{T}\Phi_s$ ,  $\text{F}_s$ ,  $\text{PN}_s$  сильно противоречит  $s$ , имеет один вход и один выход

$\text{t}_{(q)} \in \text{Z}_{\prod_{s'} \text{P}_s \text{PN}_{s'}}$  является конечным автоматом (т.е. каждый переход связан,  $\text{Z}_{\text{t}_{(q)} \text{t}_{(q)}} \subseteq \text{Z}_{\prod_{s'} \text{P}_s \text{PN}_{s'}}$ ) и для каждого  $\text{Z}_{\text{t}_{(q)} \text{F}_s}$  и  $\text{t}_{(q)}$

Это определение соответствует определению, приведенному в [DE95]. Подсеть  $\text{PN}_{s'}$ , который удовлетворяет требованиям, указанным в Определении 10, называется S-компонентом.

является сильно связанным конечным автоматом, таким, что для каждого места: если

(выходное) место перехода in

$\text{t} \in \text{Z}_{\text{PN}_s}$ , тогда

$\text{q}$  также является местом ввода (вывода) in

$\text{PN}_s$ , который  $\text{PN}_s$

$\text{q}$  является входом

$\text{t}$

$\text{PN}_s$ .

WF-сети, показанные на рис. 15 и рис. 24, являются S-охватываемыми. WF-сеть, показанная на рис. 26, не является S-охватываемой, потому что с5 отсутствует ни в одном S-компоненте. Следующая теорема показывает, что S-покрываемость является обобщением свойства свободного выбора и хорошо структурированности.

**Теорема 2** Надежная WF-сеть со свободным выбором является S-покрываемой. Надежная, хорошо структурированная WF-сеть является S-покрываемой.

**Доказательство.**

Смотрите [Aal96a].

Все надежные WF-сети, представленные в этой статье, являются S-coverable. Каждая S-coverable WF-сеть безопасна. Две WF-сети, которые не являются надежными, см. рис. 26 и рис. уре 20, не являются S-охватываемыми. Эти примеры показывают, что существует высокая корреляция между S-покрываемостью и надежностью. Похоже, что S-охватываемость является одним из основных требований, которым должно удовлетворять определение любого рабочего процесса. С формальной точки зрения, можно построить WF-сети, которые являются надежными, но не S- покрываемыми. Как правило, эти сети содержат места, которые не ограничивают запуск перехода, но которых нет ни в одном S-компоненте. (Смотрите, например, рисунок 65 в [Rei85].) С практической точки зрения, этих WF-сетей следует избегать. WF-сети, которые не являются S-покрываемыми, трудно интерпретировать, потому что структурные и динамические свойства не совпадают. Например, эти сети могут быть активными и ограниченными, но не структурно ограниченными. Нет практической необходимости в использовании конструкций, которые нарушают свойство S-покрываемости. Следовательно, мы считаем S-покрываемость основным требованием, которому должна удовлетворять любая WF-сеть.

S-покрываемость может быть проверена за полиномиальное время. К сожалению, в общем случае невозможно проверить надежность S-охватываемой WF-сети за полиномиальное время. Сложность определения надежности для WF-сети с S-покрытием зависит от PSPACE. Для большинства приложений это не является реальной проблемой. В большинстве случаев количество задач в одном определении процесса рабочего процесса меньше 100, а количество состояний меньше 200 000. Инструменты, использующие стандартные методы, такие как построение графа со- проверяемости, не имеют проблем с выполнением этих определений рабочего процесса.

## 5.5 Краткие сведения

Три структурные характеристики (свободный выбор, хорошо структурированный и S-охватываемый) оказываются очень полезными для анализа определений процессов workflow. Возможность S-покрытия является желательным свойством, которому должно удовлетворять любое определение рабочего процесса. Конструкции, нарушающие S-покрытие, могут быть легко обнаружены, и могут быть созданы инструменты, помогающие

разработчик для построения S-покрываемой WF-сети. S-покрываемость является обобщением хорошо структурированности и свойства свободного выбора (теорема 2). Как хорошо структурированная функциональность, так и свойство свободного выбора также соответствуют желаемым свойствам рабочего процесса. WF-сеть, удовлетворяющая хотя бы одному из этих двух свойств, может быть проанализирована очень эффективно. Однако мы показали, что существуют рабочие процессы, которые не основаны на свободном выборе и плохо структурированы. Рассмотрим, например, рисунок 15. Тот факт, что эта задача обрабатывает жалобу проверяет, есть ли токен в *c5* предотвращает свободный выбор или хорошо структурированную WF-сеть. Хотя это очень разумный рабочий процесс, большинство систем управления рабочими процессами не поддерживают такую продвинутую схему маршрутизации. Даже если кто-то может использовать рабочие процессы, основанные на состоянии (например, COSA), допускающие конструкции, которые нарушают хорошо структурированность и свойство свободного выбора, тогда структурные характеристики все еще полезны. Если WF-сеть не имеет свободного выбора или плохо структурирована, следует найти источник, который нарушает одно из этих свойств, и проверить, действительно ли необходимо использовать несвободную или плохо структурированную конструкцию. Если действительно необходима несвободная или плохо структурированная конструкция, то корректность конструкции следует перепроверить дважды, поскольку она является потенциальным источником ошибок.

## 6 Набор инструментов

Управление рабочими процессами - очень полезная область применения сетей Петри. С одной стороны, сети Петри позволяют проектировать сложные рабочие процессы с использованием передовых конструкций маршрутизации. С другой стороны, для проверки правильности определения рабочего процесса могут быть использованы мощные методы анализа. К сожалению, большинство систем управления рабочими потоками не основаны на сетях Петри. Есть всего несколько продуктов, которые используют сети Петри в качестве языка проектирования, например, COSA (Software Ley / COSA Solutions, Пуллхайм, Германия), INCOME (Promatis, Карлсбад, Германия) и LEU (Vebacom, Бохум, Германия).

У нас большой практический опыт применения COSA. COSA - один из ведущих продуктов на голландском рынке документооборота. Система управления документооборотом COSA состоит из ряда компонентов:

### Сетевой редактор COSA (CONE)

CONE - это инструмент для разработки определений процессов документооборота. На рисунке 28 показано окно

### CONE. Пользовательский редактор COSA (COUE)

COUE используется для классификации ресурсов с точки зрения ролей и организационных единиц.

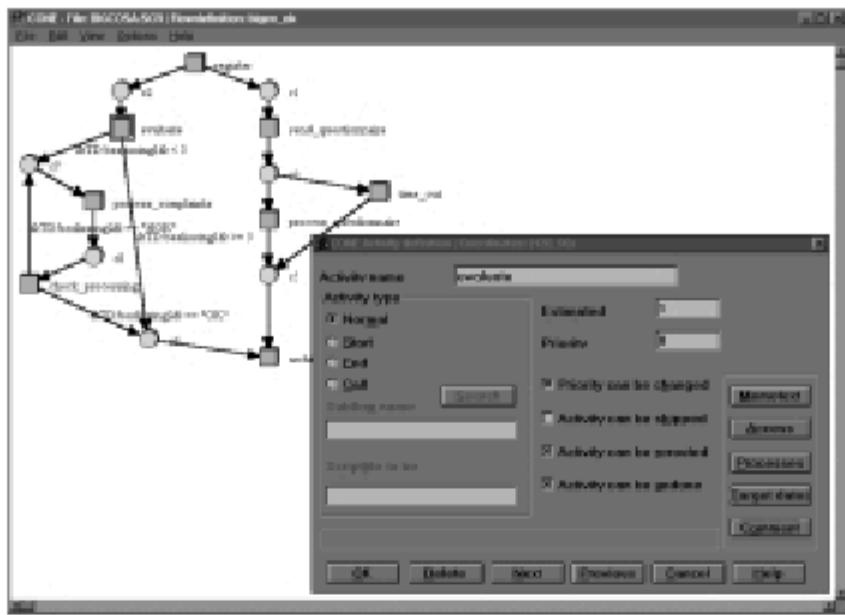


Рисунок 28: Определение процесса COSA workflow, выполненное с помощью CONE.

#### *COSA Simulator (COSI)*

COSI - это примитивный инструмент моделирования, который можно использовать для проверки рабочего процесса определения процессов.

#### *COSA Runtime Service (CORS)*

CORS - это служба внедрения рабочего процесса, которая состоит из одного или нескольких движков рабочего потока.

#### *COSA MemoBox (COMB)*

COMB - это клиентское приложение для рабочих процессов (in-basket), которое предлагает рабочие элементы конечному пользователю.

#### *Индикатор сетевого состояния COSA (COND)*

COND - это расширение COMB, которое позволяет конечному пользователю видеть текущее состояние обращения.

#### *Администратор COSA (COAD)*

COAD - это инструмент администрирования и мониторинга, который может использоваться для обработки сбоев, внесения изменений, обнаружения проблем и сбора управленческой информации.

COSA основан на сетях Петри: CONE использует сети Петри в качестве языка проектирования, COSI и COND визуализируют рабочий процесс в терминах сети Петри, а CORS использует правила запуска для запуска рабочего процесса.

COSA позволяет моделировать и внедрять сложные рабочие процессы, которые используют продвинутые конструкции маршрутизации. Однако COSA не поддерживает верификацию, она не допускает детального моделирования, и ее сложно использовать пользователям, которые просто интересуются моделированием бизнес-процессов. Для решения этих проблем мы используем COSA в сочетании с тремя другими продуктами: *Protos*, *ExSpect* и *Woflan*. Эти продукты могут взаимодействовать друг с другом. На рисунке 29 показаны отношения между COSA, Protos, ExSpect и Woflan.

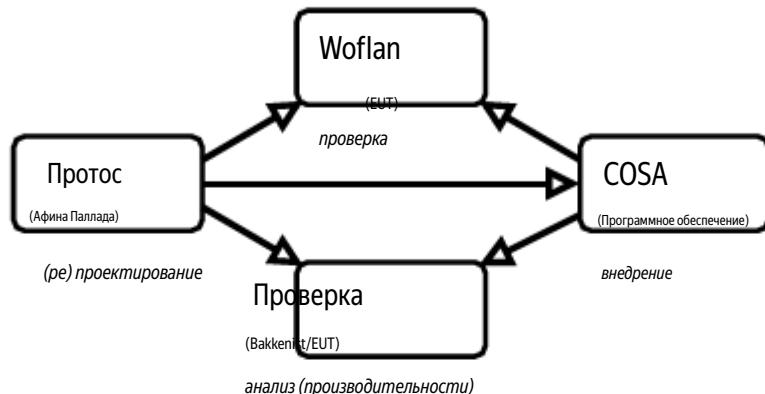


Рисунок 29: Интегрированный набор инструментов для проектирования, анализа и внедрения рабочих процессов.

Protos (Афина Паллада, Плазмолен, Нидерланды) - это так называемый BPR-инструмент. Protos поддерживает усилия по реинжинирингу бизнес-процессов (BPR) и может использоваться для управления определением и анализом бизнес-процессов ([Pal97]). Инструмент очень прост в использовании и основан на сетях Петри. Для облегчения моделирования простых рабочих процессов пользователями, не знакомыми с сетями Петри, можно абстрагироваться от состояний. Другим примером инструмента BPR, основанного на сетях Петри, является Structware/BusinessSpecs (IvyTeam, Цуг, Швейцария). Этот инструмент может использоваться для моделирования бизнес-процессов, которые могут быть экспортированы в Staffware (Staffware, Лондон, Великобритания). Staffware - это один из ведущих продуктов для документооборота в мире.

ExSpect (Bakkenist Management Consultants, Димен, Нидерланды) может быть использован для моделирования процессов, смоделированных в терминах высокогуровневых сетей Петри ([Bak96]). (Первые версии ExSpect были разработаны сотрудниками факультета математики и вычислительной техники Технологического университета Эйндховена.) Рабочие процессы, смоделированные с помощью Protos или COSA, могут быть перенесены в ExSpect. Во время моделирования рабочего процесса с помощью ExSpect, рабочий процесс анимируется и измеряются ключевые показатели производительности (например, средняя пропускная способность, время и интенсивность занятости).

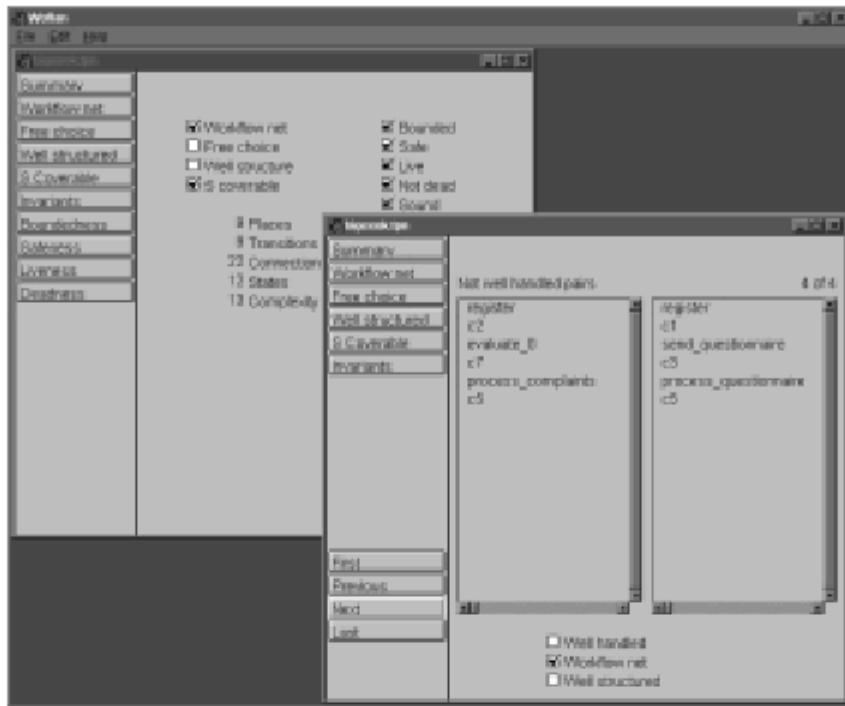


Рисунок 30: Одно из многочисленных окон анализа в Woflan.

Woflan (Технологический университет Эйндховена, Эйндховен, Нидерланды) - это анализатор рабочих процессов на основе сети Петри ([HVA97, AHV97]). Woflan (WOrkFLow AN- анализатор рабочего процесса) может использоваться для проверки правильности рабочего процесса.

Инструмент не зависит от системы управления рабочим процессом, т.е. Предоставляется ряд функций импорта для загрузки сценариев рабочего процесса в Woflan (например, из COSA и Protos). Woflan использует стандартные методы анализа, основанные на сети Петри. Однако результаты анализа представлены таким образом, чтобы конечные пользователи могли понять результаты работы Woflan. Более того, Woflan помогает конечному пользователю исправить ошибочный рабочий процесс . Одним из центральных вопросов, который анализирует Woflan, является качество надежности. Недостаточно определить, является ли определение рабочего процесса надежным. Во многих случаях необходимо удовлетворить дополнительные требования. Более того, если определение рабочего процесса неверно, то пользователю следует руководствоваться при обнаружении источника ошибки и оказывать поддержку для устранения ошибки. Вот почему Woflan предлагает большой выбор методов анализа, включая методы, обсуждаемые в этой статье. Например, Woflan обнаруживает конструкции, которые нарушают свойство свободного выбора, хорошо структурируемость или S-покрываемость, вычисляет инварианты, обнаруживает неограниченные места, и сообщает о мертвых переходах.

## 7 Заключение

В этой статье мы обсудили применение сетей Петри к области рабочих процессов.

Управление рабочими процессами оказывается прикладной областью, которая могла бы извлечь выгоду из возможностей сетей Петри. Существует по крайней мере три веские причины для использования сетей Петри для моделирования и анализа рабочих процессов ([Aal96b]):

### *Формальная семантика, несмотря на графический характер*

С одной стороны, сети Петри - это графический язык, который позволяет моделировать примитивы рабочего процесса, определенные WfMC. С другой стороны, семантика сетей Петри (включая большинство расширений) была определена формально. Многие из доступных сегодня систем управления рабочими процессами предоставляют специальные конструкции для моделирования процедур рабочего процесса. Более того, существуют системы управления рабочими процессами, которые накладывают ограничения на примитивы рабочего процесса, показанные на рисунке 12. Некоторые системы управления рабочими процессами также предоставляют экзотические конструкции, семантика которых не ясна на 100%. Из-за этих проблем лучше использовать устоявшийся язык проектирования с формальной семантикой.

### *Основанный на состоянии, а не на событии*

В отличие от многих других методов моделирования процессов, состояние дела может быть явно смоделировано в сети Петри. Методы моделирования процессов, варьирующиеся от неформальных, таких как диаграммы потоков данных, до формальных, таких как алгебра процессов, основаны на событиях, т.е. переходы моделируются явно, а состояния между последующими переходами моделируются неявно.

Современные системы управления рабочими процессами обычно основаны на событиях, т. е. Задачи моделируются явно, а состояния между последующими задачами подавляются. Различие между описанием, основанным на событиях, и описанием, основанным на состоянии, кажется очень тонким, но примеры в этой статье (например Рисунок 16) показывают, что это имеет первостепенное значение для моделирования рабочего процесса. Например, в большинстве систем управления рабочими процессами, которые абстрагируются от состояний, невозможно использовать неявное ИЛИ-разделение.

### *Обилие методов анализа*

Сети Петри отличаются доступностью многих методов анализа. Очевидно, что это большое преимущество сетей Петри. В этой статье мы сосредоточились на проверке рабочих процессов. Мы видели, что технологии анализа на основе сети Петри могут быть использованы для определения правильности определения рабочего процесса. Доступность этих методов иллюстрирует, что теория сетей Петри может быть использована для добавления мощных аналитических возможностей в следующее поколение систем управления рабочими процессами.

Сегодняшняя ситуация в отношении программного обеспечения для управления рабочими процессами сравнима с ситуацией в отношении программного обеспечения для управления базами данных в начале семидесятых годов. В начале семидесятых большинство пионеров в области СУБД использовали свои собственные специальные концепции. Эта ситуация беспорядка и отсутствия понимания привела к появлению непостижимого набора СУБД. Тем не менее, новые стан- Дарты например, реляционной модели данных [Cod70] и "сущность-связь" МО- дель [Che76] приведет к общей формальной основой для многих СУБД. В результате использование этих СУБД расширилось. Существует много общего между современными системами управления документооборотом и СУБД начала семидесятых. Несмотря на усилия Коалиции по управлению рабочими процессами, реальный концептуальный стандарт отсутствует. В результате многие организации неохотно используют существующее программное обеспечение для управления рабочими процессами . По нашему мнению, сети Петри представляют собой хорошую основу для стандартизации. Мы только что привели три веские причины для использования системы управления рабочими процессами на основе сети Петри . Вдохновленные практическим опытом, мы пришли к пониманию того, что многие особенности формализма сети Петри полезны в контексте управления рабочим процессом .

## Список литературы

- [Aal94] W.M.P. van der Aalst. Применение сетей Петри в промышленности. *Компьютеры в промышленности*, 25(1):45-54, 1994. W.M.P.
- [Aal95] van der Aalst. Класс сетей Петри для моделирования и анализа бизнес-процессов.. Отчеты по вычислительной технике 95/26, Эйндховенский технологический университет, Эйндховен, 1995.
- [Aal96a] В.М.П. ван дер Алст. Структурные характеристики надежных сетей рабочего потока. Отчеты по вычислительной технике 96/23, Технологический университет Эйндховена, Эйндховен, 1996. В.М.П. ван дер Алст. Три веские причины для использования системы управления рабочими процессами на основе сети
- [Aal96b] Петри . В S. Navathe и T. Wakayama, под редакцией авторов, *Материалы Международной рабочей конференции по информации и интеграции процессов на предприятиях (IPC'96)*, страницы 179-201, Кэмбридж, Массачусетс, ноябрь 1996 г.
- У.М.П. ван дер Алст. Проверка сетей документооборота. В P. Azema и G. Бальбо, редакторы, *Применение и теория сетей Петри*, 1997, том 1248 из Конспекты лекций по информатике, страницы 407-426. Springer-Verlag, Berlin, 1997.

- [AX97] В.М.П. ван дер Алст и К.М. ван Хи.  
*Modellen, Methoden en Systemen (in Dutch)*.  
 Шунховен, 1997.
- Управление рабочим процессом:  
 Академическая служба,
- [AHV97] В.М.П. ван дер Алст, Д. Хаушильдт и Х.М.В. Вербик. Инструмент для анализа рабочих процессов на основе сети Петри. В Б. Фарвере, Д. Молдте и М.О. Стере, редакторах, *Труды по сетям Петри в системной инженерии (PNSE'97)*, страницы 78-90, Гамбург, сентябрь 1997 г. Хамбургский университет (FBI-HH-B-205/97).
- [Bak96] Консультанты по менеджменту из баккенистов. *Руководство пользователя ExSpect 5.0*, 1996.
- [BCD95] К. Баркауи, Ж.М. Куврер и К. Дютейе. О живучести в бывших несамоуправляющихся сетях, как правило. В Г. Де Микелис и М. Диас, редакторы, *Применение и теория сетей Петри, 1995*, том 935 из Конспекты лекций по информатике, страницы 25-44. Springer-Verlag, Berlin, 1995.
- [Bes87] E. Best. Структурная теория сетей Петри: перерыв в свободном выборе. В W. Брауэр, У. Рейсиг и Г. Розенберг, редакторы, *Достижения в области сетей Петри 1986, Часть I: Сети Петри, центральные модели и их свойства*, том 254 из Конспекты лекций по информатике, страницы 168-206. Springer-Verlag, Берлин, 1987.
- [CEP93] А. Ченг, Дж. Эспарза и Дж. Палсберг. Результаты по сложности для 1-безопасных сетей. В R.K. Shyamasundar, редактор, *Основы технологии программного обеспечения огу и теоретической информатики*, том 761 из Конспекты лекций по информатике, страницы 326-337. Springer-Verlag, Berlin, 1993.
- [Che76] P.R. Chen. Модель "Сущность-отношения": на пути к единому представлению данных. *Транзакции ACM в системах баз данных*, 1: 9-36, январь 1976 г.
- [Cod70] Э.Ф. Кодд. Реляционная модель для больших совместно используемых банков данных. *Коммуникации ACM*, 13(6): 377-387, июнь 1970.
- [Des95] Дж. Десел и Дж. Эспарза. *Сети Петри свободного выбора*, том 40 из *Самостоительные трактаты в теоретической информатике*. Издательство Кембриджского университета, Кембридж, 1995.
- Дж. Десел. Доказательство теоремы о ранге для расширенных сетей со свободным выбором. В К. Дженсен, редактор, *Применение и теория сетей Петри 1992*, том 616 из Конспекты лекций по информатике, страницы 134-153. Springer-Verlag, Berlin, 1992.

- [EKR95] Г. Розенберге. системы документооборота. В Н. Комстоке и К. Эллисе, редакторах, *Конференция по организационным вычислительным системам*, страницы 10-21. ACM SIGOIS, ACM, август 1995 г. Милпитас, Калифорния.
- [EN93] К.А. Эллис и Г.Дж. Натт. Моделирование и внедрение систем документооборота. темs. В М. Аджмоне Марсане, редактор, *Применение и теория сетей Петри 1993*, том 691 из *Конспектов лекций по информатике*, страницы 1-16. Springer-Verlag, Берлин, 1993.
- [ES90] Дж. Эспарза и М. Сильва. Схемы, мосты и сетки. В Г. Розенберг, редактор, *Достижения в области сетей Петри, 1990*, том 483 из *Конспекты лекций по информатике*, страницы 210-242. Springer-Verlag, Berlin, 1990.
- [Esp90] J. Esparza. Правила синтеза сетей Петри и то, как они могут привести к новым результатам. В J.C.M. Baeten и J.W. Klop, редакторы, *Материалы CONCUR 1990*, том 458 из *Конспекты лекций по информатике*, страницы 182-198. Springer-Verlag, Berlin, 1990.
- [GCEV72] К. Гостеллоу, В. Серф, Г. Эстрин и С. Волански. Надлежащее завершение потока управления в программах, включающих параллельные процессы. *ACM Sigplan*, 7(11):15-27, 1972.
- [Hac72] Взлом М.Н.Т.. Производственные схемы анализа с помощью сетей Петри. мастерса , Массачусетский технологический институт, Кембридж, Массачусетс, 1972. Диссертация
- [Hee94] К.М. ван Хи. *Разработка информационных систем: формальный подход..* Издательство Кембриджского университета, 1994. К. Хейс
- [HL91] и К. Лавери. Программное обеспечение для управления рабочими процессами: возможности для бизнеса. Технический отчет, Ovum Ltd, Лондон, 1991. Д. Хаушильдт, Х.М.В. Вербик и В.М.П. ван дер Алст.
- [HVA97] WOFLAN: анализатор рабочего процесса на основе сети Петри. Научные отчеты 97/12, Технологический университет Эйндховена, Эйндховен, 1997. Computing
- [Jen96] К. Дженсен. *Цветные сети Петри. Основные концепции, методы анализа и практическое использование.* Монографии EATCS по теоретической информатике. Springer-Verlag, Berlin, 1996.

- [Joo94] С. Джостен. Моделирование триггеров для анализа рабочего процесса. В Г. Крут и А. Бенцур, редакторы, *Proceedings CON'94: Управление рабочими процессами, проблемы, парадигмы и продукты*, страницы 236-247, Вена, октябрь 1994г.
- [Kou95] Т.М. Кулупулос. *Императив рабочего процесса*. Van Nostrand Рейнхольд, Нью-Йорк, 1995. П. Лоуренс, редактор.
- [Law97] *Справочник по рабочим процессам 1997, Коалиция по управлению рабочими процессами*. Джон Уайли и сыновья, Нью-Йорк, 1997.
- [MBea95] М. Аджмон Марсан, Г. Бальбо, Г. Конте и др. *Моделирование с Покол-  
ализed стохастические сети Петри*. Серия Уайли в параллельных вычислений. Вайли, Нью-Йорк, 1995. [MEM94]
- Г. Де Микелис, К. Эллис и Г. Мемми, редакторы. *Материалы  
второго семинара по совместной работе с компьютерной поддержкой,  
сетям Петри и связанным с ними формализмом*. Сарагоса, Испания, июнь 1994 г. Т. Мурата. Сети Петри: свойства, анализ и приложения. *Разработки стандарта IEEE*, 77 (4): 541-580, апрель 1989.
- [Pal97] Афина Паллада. *Руководство пользователя Protos*. Pallas Athena BV, Плазмолен, Нидерланды, 1997.
- [Pet62] С.А. Петри. *Kommunikation mit Automaten*. PhD thesis, Institut fur instrumentelle Mathematik, Bonn, 1962.
- [Rei85] W. Reisig. *Сети Петри: введение*, том 4 из *Монографий по теоретической информатике: серия EATCS*. Springer-Verlag, Berlin, 1985.
- [SH95] П.А. Штрауб и К. Уртадо. Свойство простого управления моделями бизнес-процессов. На XV Международной конференции Чилийского общества компьютерных наук, 1995. Программное обеспечение.
- [SL96] *Руководство пользователя COSA*. Программное обеспечение-Ley GmbH, Пуллхайм, Германия, 1996. Р. Валк. Безграничное поведение и честность. У. Брауэра, У. Рейсига и Г. Розенберг, редакторы, *Достижения в области сетей Петри, 1986, Часть I: Сети Петри, центральные модели и их свойства*, том 254 из *Конспекты лекций по информатике*, страницы 377-396. Springer-Verlag, Berlin, 1987.

- [WFM96] WFMC. Терминология и глоссарий Коалиции по управлению рабочими процессами (WFMC-TC-1011). Технический отчет, Сотрудничество по управлению рабочими процессами, Брюссель, 1996.
- [Wil89] Р. Уайлд. *Управление производством и операциями: принципы и методы* Касселл, Лондон, 1989. М. Вольф
- [WR96] и У. Реймер, редакторы. *Материалы Международной конференции по практическим аспектам управления знаниями (РАКМ'96), Семинар по адаптивному документообороту*, Базель, Швейцария, октябрь 1996 г.