

doi: 10.36724/2409-5419-2021-13-6-60-68

ИНТЕГРАЦИЯ СРЕДСТВ АНАЛИЗА И ИСПОЛНЕНИЯ БИЗНЕС-ПРОЦЕССОВ НА ОСНОВЕ БЛОЧНОГО ПОДХОДА

ВАСИЛЬЕВ

Николай Владимирович¹

ЯШИН

Александр Иванович²

ДОВЖИКОВ

Сергей Николаевич³

АННОТАЦИЯ

Введение: ядром современных информационных систем управления предприятием являются машины исполнения бизнес-процессов. Подавляющее их большинство используют язык описания BPMN2 (например JBPM, Activiti, Camunda). Несмотря на принятие стандарта, поддержка BPMN редакторами платформо-ориентирована и обладает ограниченными возможностями по переносу описания. Для создания и поддержания BPMN описаний в актуальном состоянии бизнес-аналитик помимо редакторов использует целый набор вспомогательных инструментов, включающих средства верификации (например WoPeD) и анализа производительности (Bizagi, BIMP). Свои особенности вносит привлечение для выявления и актуализации бизнес-процессов методологии глубинного анализа процессов (Process Mining), использующей свой набор инструментов (Disco, ProM), поддержка BPMN в которых ограничена. Как следствие интеграция инструментов анализа и средств исполнения бизнес-процессов остается недостаточной, что существенно замедляет внедрение и актуализацию систем управления предприятием. **Цель исследования:** на базе общей методологии обработки журналов, используемой Process Mining, показать, что блочно-структурированный подход на основе деревьев процессов может быть универсальным формализмом на всех этапах (реконструкция, анализ качества, анализ временных и стоимостных параметров). **Методы:** теория сетей Петри, дискретная математика, математическая логика, теория алгоритмов. **Результаты:** описана схема интегрированного анализа бизнес-процессов. Предложена и реализована модель динамики деревьев, которая является основой для реализации машины исполнения бизнес-процессов. На основе модели динамики предложен алгоритм построения графа достижимости маркировок дерева процессов. Модифицированы алгоритмы оценки соответствия и точности для их применения с деревьями процессов. В рамках методологии аналитической оценки параметров производительности блочных бизнес-процессов получена формула оценки времени завершения циклического блока с несколькими вариантами доработки. Произведено экспериментальное исследование эффективности разработанного решения путем натурального моделирования схемы интегрированного анализа и предложенных методов и его сравнение с промышленными инструментами (pm4py) по критериям точности и соответствия реконструкции. Полученные экспериментальные результаты позволяют говорить о соизмеримых с известными инструментами характеристиках. Это позволяет уменьшить необходимость использования внешних инструментов, повышая оперативность внесения изменений. **Практическая значимость:** приведенные в работе идеи были реализованы в разработанной авторами виртуальной машине Onegine и прошли апробацию в ряде опытных конструкторских работ. Описанные в работе результаты могут служить основой для создания более эффективных систем управления предприятием.

Сведения об авторах:

¹ к.т.н., начальник сектора, ПАО "Интелтех", г. Санкт-Петербург, Россия, gandvik1984@gmail.com

² д.т.н., советник генерального директора ПАО "Интелтех", г. Санкт-Петербург, Россия, a_yashin@inbox.ru

³ инженер, ПАО "Интелтех", г. Санкт-Петербург, Россия, sergei.dovzhikov@gmail.com

КЛЮЧЕВЫЕ СЛОВА: деревья процессов, верификация, машины исполнения бизнес-процессов, сети Петри, производительность бизнес-процессов.

Для цитирования: Васильев Н.В., Яшин А.И., Довжиков С.Н. Интеграция средств анализа и исполнения бизнес-процессов на основе блочного подхода // Научные технологии в космических исследованиях Земли. 2021. Т. 13. № 6. С. 60–68. doi: 10.36724/2409-5419-2021-13-6-60-68

Введение

Платформы корпоративных информационных систем, примерами которых являются СЭД и ERP – системы, обладают развитыми инструментами адаптации к особенностям объектов автоматизации. Модификация прикладной логики этих систем обычно осуществляется через описываемые в графической нотации BPMN бизнес-процессы. Методологически [1, 2] создание схем бизнес-процессов осуществляется посредством формализации нормативных документов и опроса сотрудников на рабочих местах. Полученное описание опционально подвергается имитационному моделированию и развертывается в корпоративной информационной системе. На последующих этапах оптимизации и внедрения бизнес-процессы часто требуют многочисленных уточнений. Задача осложняется тем, что поддержка BPMN в инструментах часто неполная и перенос описаний между инструментами проектирования, моделирования и исполнения затруднен. Как следствие, аналитик при исследовании и развертывании бизнес-процесса вынужден создавать специфические версии одной и той же модели.

Привлечение методологии глубинного анализа процессов (process mining) позволяют повысить эффективность создания бизнес-процессов [3, 4]. Методология предполагает, что на предприятии ставится система с минимально необходимым набором процессов. Пользователи выполняют в системе привычные действия с минимальной долей автоматизации, фактически в ручном режиме. Действия пользователей заносятся в журналы информационных систем, обработав которые аналитик может принять рациональное решение об организации бизнес-процесса.

Но и в данной методологии вновь имеет место многообразие моделей, специфичных для различных алгоритмов (например, D/F – графы процессов, сети Петри, нечеткие схемы). Как результат бизнес-аналитик должен и в этом случае выполнять значительное количество ручных операций по преобразованию моделей.

Ключ к решению описанных трудностей [5], [6] – унификация моделей анализа и исполнения бизнес-процессов. Перспективно [7] применение высокоуровневого блочно-структурированного подхода к описанию на основе деревьев процессов. За счет простоты и корректности на уровне синтаксиса, данный подход может служить единым формализмом, связывающим основные этапы жизненного цикла бизнес-процессов.

Данная работа посвящена решению некоторых прикладных и теоретических вопросов, обеспечивающих практическую реализуемость сквозного подхода к описанию, анализу и исполнению бизнес-процессов, задаваемых при помощи деревьев. Описываемые подходы были реализованы в ряде изделий. В работе описывается модифицированная авторами схема интегрированного анализа и исполнения бизнес-процессов. Для формализации исполнения предложена модель динамики деревьев процессов. Далее приведены и обоснованы модифицированные авторами алгоритмы оценки соответствия и точности реконструированного дерева. В следующих разделах приводится полученная авторами методика прогнозирования времени выполнения процессов.

В конце работы приведены результаты натурного моделирования и сравнение результатов реконструкции с существующими инструментами process mining (pm4py) на основе метрик точности и соответствия.

Схема интегрированного анализа процессов

В работе для реконструкции схем высокоуровневых процессов используется журнал выполнения уже развернутых в информационной системе элементарных (базовых) процессов. Например, в системах электронного документооборота базовыми действиями будут подписание документа цифровой подписью, создание дочернего поручения, просмотр документа, вынесение резолюции и пр. Базовые процессы ассоциированы с определенным типом объектов и состоят из единственного действия изменения состояния объекта, характеризуются ролью исполнителя и набором вспомогательных атрибутов, отображаемых на форме. Общая схема интегрированного анализа представлена на рис. 1. За счет выделения относительно неизменной модели элементарных операций, такой подход позволяет произвести оперативное изменение развернутых в корпоративной информационной системе бизнес-процессов, которыми для приведенного примера электронного документооборота будут маршруты движения документов в организации. По мере заполнения журнала элементарными действиями становится возможно средствами Process Mining исследовать логику их связей в процессе выполнения. Как только развернутый бизнес-процесс перестает удовлетворять потребностям организации, пользователи вновь переходят к базовым действиям и процесс реконструкции повторяется.

В корпоративных информационных системах каждый тип объектов (вид документа, запись справочника и пр.) характеризуется своим набором специфичных бизнес-процессов. Поэтому возможно разбиение общесистемного журнала по типам объектов (далее назовем его объектом-селектором).



Рис. 1. Схема интегрированного анализа процессов

Каждый объект-селектор характеризуется *жизненным циклом*, который описывает его поведение от момента создания до уничтожения. Жизненный цикл может быть описан в форме графа, в котором вершинами являются стадии, а ребрами – переходы между стадиями.

В предлагаемом подходе реконструируется единая схема бизнес-процесса, отражающая все стадии жизненного цикла объекта-селектора от создания до уничтожения. Задача разбиения единой схемы процесса на подпроцессы ложится на аналитика. Методологически удобной является декомпозиция дерева процесса по стадиям жизненного цикла объектов-селекторов. Точность и соответствие модели (см. раздел 4) процесса также удобнее оценивать сразу для всего дерева поведения объекта-селектора. Указанные интегральные характеристики позволяют осуществлять непрерывный мониторинг периодически формируемого журнала и эксплуатировать совокупности бизнес-процессов.

Оценка временных характеристик созданных процессов (см. раздел 4) выполняется на последнем этапе, так как не влияет на процесс реконструкции и имеет значение при оптимизации логики и прогнозировании сроков завершения экземпляров бизнес-процессов.

Как было указано, для заполнения журнала в реализованной авторами машине исполнения Onengine [7] выделяется специальный вид базовых бизнес-процессов. Реконструкция модели процесса была осуществлена алгоритмом InductiveMiner [8-10]. В разделе 2 описывается модель динамики деревьев процессов, используемая в разделе 3 для модификации алгоритмов структурной оценки качества реконструкции. Приведенный в следующем разделе формализм также используется в разделе 4 при декомпозиции модели процесса для оценки времени выполнения.

Модель динамики деревьев процессов

Логической основой для BPMN является подкласс сетей Петри, носящих название WF-сетей [1, 2]. Как и в случае сетей Петри, BPMN не боится от возможных ошибок: тупиков, активных блокировок и мертвых переходов.

Интуитивное свойство корректности процесса, выражающееся в том, что бизнес-процесс должен стартовать с одного маркера в начальном состоянии, заканчиваться одним маркером в последнем состоянии и не содержать неисполняемых элементов составляют суть концепции бездефектности [1, 2].

Проблема возможности внесения разработчиками логических ошибок разрешается при блочно-структурированном задании бизнес-процесса в виде дерева. Данный подход, рассмотренный в работах [8-10], позволяет создавать гарантированно бездефектное описание. Однако деревья процессов пока используются главным образом в алгоритмах ProcessMining. Применение этого подхода в машинах исполнения, а также на этапе оценки адекватности исполняемого бизнес-процесса требует разработки более специфичной, чем в сетях Петри модели динамики. Требуется выработать правила срабатывания дерева процессов, а также модифицировать основные понятия и методы анализа, сформулированные для сетей Петри [11].

Теоретические основы формализма деревьев процессов следуют из доказанной теоремы [1] о том, что любой бездефектный процесс можно разбить на блоки, каждый из которых имеет на входе и выходе один маркер. Если каждый из таких блоков бездефектен, то их можно объединить при по-

мощи операторов в процесс более высокого уровня. Формализм дополняет параллельный (\wedge) и условный (\times) операторы композиции блоков циклическим ($@$) и последовательным (\rightarrow) способами. Листовыми узлами дерева являются выполняемые действия.

Заметим, что исходя из теоремы о бездефектности блочно-структурированных процессов [1,2], между родительским и дочерним узлом всегда будет передаваться только один маркер. То есть любой узел как формальное представление блока получает от родительского узла единственный маркер. После выполнения собственных дочерних узлов, маркер передается обратно в родительский узел. Перемещение маркера по поддеревьям в зависимости от типа узла показано на рисунке 2.

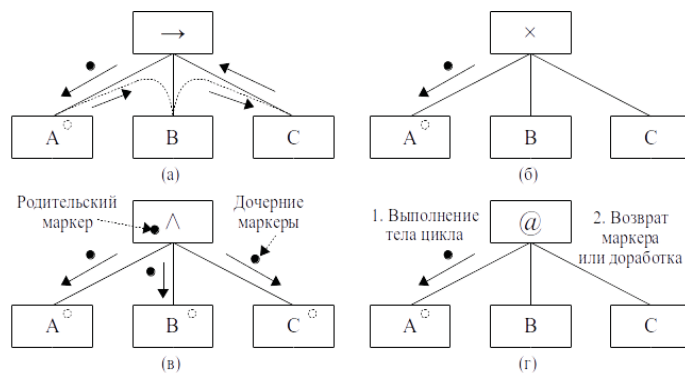


Рис. 2. Обход маркерами поддеревьев последовательного (\rightarrow), условного (\times), параллельного (\wedge) и циклического ($@$) узлов

В случае последовательного узла (\rightarrow) единственный маркер поочередно передается в поддеревья, соответствующие дочерним узлам. Такое исполнение соответствует обходу в глубину. В условном узле (\times) единственный маркер передается в выбранное дочернее поддерево. При входе маркера в узел параллельного выполнения (\wedge) происходит создание дочерних маркеров, каждый из которых передается в соответствующее поддерево. Исходный (родительский) маркер ожидает момента возврата маркеров от дочерних поддеревьев. После получения маркеров от всех дочерних поддеревьев происходит их уничтожение, а родительский маркер передается узлу-предку.

Указанную схему синхронизации можно упростить учитывая бездефектность дерева процессов. Будем считать, что маркер передается из параллельного узла в родительский в случае, если от дочернего узла был получен маркер, а в других узлах-потомках больше не осталось маркеров. Узел циклического выполнения ($@$) содержит тело цикла A, возможный вариант доработки конечного результата B и выход из цикла C. При обходе единственный маркер передается в первое по порядку поддерево. После его завершения поведение, фактически, аналогично исключающему узлу с поддеревьями C и B. В случае доработки маркер передается в соответствующее поддерево. По его возвращению, маркер снова передается в первое поддерево. После чего снова производится исключающий выбор между завершением или доработкой.

В процессе функционирования маркеры могут «останавливаться» только в листовых узлах, ожидая завершения дей-

ствия. Таким образом состояние процесса можно описать расположением маркеров в дочерних узлах. Будем обозначать листовые узлы дерева как p_1, p_2, \dots, p_n .

Маркировкой дерева назовем вектор вида (p_1, p_2, \dots, p_n) . Элемент p_i этого вектора указывает на присутствие маркера в узле (0 или 1). Иными словами, маркировка дерева – функция $M(p)$, сопоставляющая каждому листовому узлу дерева p число находящихся в нем маркеров:

$$M(p): P \rightarrow \{0,1\}$$

Листовой узел p *маркирован (marked)*, если содержит маркер. Маркированный узел может сработать, вызывая смену маркировки. Сработавший в маркировке M_1 листовой узел a порождает множество новых маркировок $\{M_i\}$ перемещая маркеры между листовыми узлами. В случае узлов последовательного (\rightarrow) и параллельного (\wedge) операторов множество $\{M_i\}$ будет состоять из единственной маркировки, в то время как при срабатывании условного (\times) и циклического ($@$) узлов число порождаемых маркировок будет равно числу дочерних поддеревьев. Маркировки, входящие в множество $\{M_i\}$, будем называть «сцепленными».

Будем обозначать этот факт как $M_1 \xrightarrow{a} \{M_i\}$. Будем считать, что маркеры между узлами перемещаются мгновенно и в один и тот же момент времени может сработать только один листовой узел.

Можно выделить 4 состояния листового узла: не содержит маркера и не получает его в такте (неактивный), получает маркер в такте (активация), отдает маркер в родительский узел (срабатывание), пропускает срабатывание в такте (ожидание).

Таким образом, можно считать дерево процессов (без листовых узлов) «оператором», перемещающим маркеры между листовыми узлами. Динамику дерева процессов удобно описать в форме графа достижимости маркировок, в котором вершинами являются маркировки, а переходами – сработавшие действия листовых узлов. Формально, алгоритм графа достижимости маркировок может быть описан следующим образом:

PT – REACHABILITY – GRAPH

```

1  $(V, E, v_0) := (\{M_0\}, \emptyset, M_0)$ ;
2  $Work := \{M_0\}$ ;
3 while  $Work \neq \emptyset$ 
4   do выбрать  $M$  из  $Work$ ;
5      $Work := Work \setminus \{M\}$ ;
6     for  $a \in enabled(M)$ 
7       do  $\{M'_i\} := fire(M, a)$ ;
8         for  $M'_i \in \{M'_i\}$ 
9           do if  $M'_i \notin V$  then  $V := V \cup M'_i$ ;
10             $Work := Work \cup M'_i$ ;
11             $E := E \cup \{(M, a, M'_i)\}$ ;
12 return  $(V, E, v_0)$ 
```

В псевдокоде $enabled(M)$ – множество листовых узлов (действий), которые могут сработать в маркировке M . $fire(M, a)$ – функция, приводящая к смене маркировке при выполнении листового узла a .

Алгоритм, за исключением упомянутого недетерминизма условного и циклического шлюзов, аналогичен алгоритму построения графа достижимости сети Петри [2]. Однако, в отличие от последнего всегда корректно завершается вследствие ограниченности описываемых деревьями процессов (вытекает из теоремы о бездефектности) [1-3]. Пример дерева процессов и графа маркировок показан на рисунке 3.

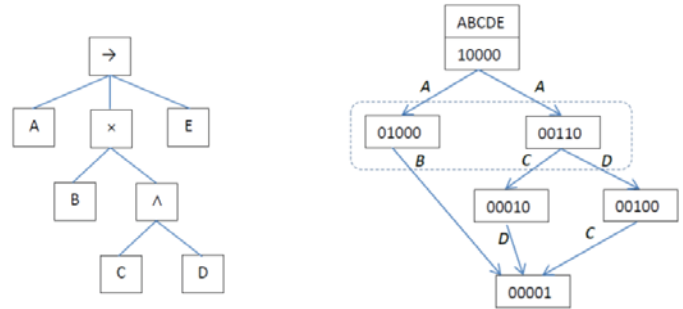


Рис. 3. Схема дерева процессов и его граф достижимости маркировок. Пунктиром показано множество «сцепленных» маркировок

В разработанном прототипе для реконструкции схемы процесса был использован алгоритм InductiveMiner [8]. Следующий раздел посвящен оценке качества произведенной реконструкции применительно к деревьям процессов.

Модель динамики деревьев процессов

Для оценки качества реконструкции моделей процессов были использованы метрики соответствия [12, 13] и точности [14].

Высокий уровень соответствия (fitness) обеспечивает возможность воспроизведения реконструированной моделью поведения в журнале событий.

Точность характеризует отсутствие избыточности реконструированной модели по отношению к журналу.

Для оценки соответствия был использован подход [12], модифицированный для деревьев процессов. Алгоритм базируется на вычислении оптимального выравнивания трасс журнала L и последовательности срабатываний листовых узлов модели процесса PT . Метрика соответствия определяется как:

$$fitness(L, PT) = 1 - \frac{fcost(L, PT)}{move(L) + |L|move(PT)} \quad (1)$$

где $fcost(L, PT)$ – минимальное суммарное расстояние Левенштейна между выравненной трассой журнала и последовательностью срабатываний модели PT ;

$move(L)$ – суммарная длина трасс журнала событий

$move(PT)$ – длина наиболее короткой трассы модели процесса PT .

Наиболее сложным в приведенном алгоритме является вычисление оптимального выравнивания трассы журнала и последовательности срабатываний модели. Алгоритм базируется на идее совместного исполнения трассы l и дерева PT .

Состояние совместного выполнения процесса и трассы описывается кортежем $(M_i(PT), M_j(l))$, где $M_i(PT)$ – текущая маркировка дерева процессов PT , $M_j(l)$ – текущее выполненное действие трассы l .

Каждое совместное состояние имеет историю выполнения, которая представляет собой выравнивание процесса и трассы вплоть до текущей маркировки процесса и трассы. Например:

$PT: \{a b c d e \wedge\}$
 $l: \{\wedge b \wedge d e f\},$

где \wedge – пропуск символа в модели PT или в трассе l .

Будем обозначать выравнивания как A_i . Каждое совместное состояние будем характеризовать выравниванием $A_i = \text{alignment}(M_i(PT), M_j(l))$.

На множестве выравниваний $\{A_i\}$ вводится функция стоимости $\text{Price}(A_i)$, равная числу столбцов в выравнивании содержащих различные символы. Для пустого множества действий $\text{Price}(\emptyset) = \infty$.

На указанной совместной маркировке могут быть заданы модифицированная функция $\text{enabled}(M_i(PT), M_j(l))$, возвращающая множество пар действий (a_{PT}, a_l) , которые могут сработать в совместном состоянии $(M_i(PT), M_j(l))$ и функция $\text{fire}(M_i(PT), M_j(l), a_{PT}, a_l)$, возвращающая новое совместное состояние при выполнении действий a_{PT}, a_l на модели процесса PT и трассе l . Допустимы следующие пары действий:

(a, a) – совместное выполнение действия a на модели и трассе;

(a, \perp) – выполнение действия a на модели и пропуск хода на трассе;

(\perp, b) – пропуск действия на модели и выполнение b на трассе.

Алгоритм вычисления оптимального выравнивания использует принцип динамического программирования. Основной структурой выступает приоритетная очередь маркировок MQ , в которую производится добавление сгенерированных совместных маркировок $(M_i(PT), M_j(l))$ и соответствующего совместному состоянию выравнивания $A_i = \text{alignment}(M_i(PT), M_j(l))$. В очереди добавленные совместные маркировки упорядочены по величине $\text{Price}(A_i)$.

Также используется вторая очередь – результатов MR , в которую помещаются финальные совместные маркировки. При запуске очередь MQ инициализируется начальными маркировками $\{(M_0, _)\}$. LL – список проанализированных маркировок. Из очереди MQ извлекается самая «хорошая» совместная маркировка (строка 4), стоимость которой меньше, чем лучшей маркировки из очереди результатов MR (условие выхода из цикла). Если извлеченная совместная маркировка – конечная, то она добавляется в очередь результатов MR (строки 6, 7). В противном случае сгенерированные и еще не проанализированные совместные маркировки добавляются в очередь результатов MR .

PT – ALIGNMENT

1 $(V, E, \{v_0\}) := (\{(M_0, _)\}, \emptyset, \{(M_0, _)\})$;
 2 $MQ := \{(M_0, _)\}, MR := \{\emptyset\}, LL := \{\emptyset\}$;
 3 **while** $\text{Price}(\text{Best}(MQ)) < \text{Price}(\text{Best}(MR))$

4 **do** $(M(PT), M(l)) = \text{Best}(MQ)$;
 5 $MQ := \text{Work} \setminus \{(M(PT), M(l))\}$;
 6 **if** $\text{enabled}(M(PT), M(l)) = \emptyset$
 7 **then** $MR := MR \cup (M(PT), M(l))$ **else**
 8 **for** $(a, b) \in \text{enabled}(M(PT), M(l))$
 9 **do** $\{(M'(PT), M'(l))\} := \text{fire}(M(PT), M(l), a, b)$;
 10 **for** $(M'(PT), M'(l)) \in \{(M'(PT), M'(l))\}$
 11 **do if** $M'(PT), M'(l) \notin LL$
 14 **then** $LL := LL \cup (M'(PT), M'(l))$
 14 $MQ := MQ \cup (M'(PT), M'(l))$
 15 **return** $\text{Best}(MR)$

Приведенный алгоритм всегда останавливается по причине конечности графа маркировок дерева процессов, как следствие бездефектности на уровне синтаксиса. Оптимальность найденного выравнивания следует из принципа динамического программирования.

Метрика точности характеризует отсутствие избыточности модели процесса по отношению к журналу. Она может быть вычислена как [14,15]:

$$\text{precision}(L, PT) = \frac{\sum_{s \in Q} \omega(s) \cdot |e_x(s)|}{\sum_{s \in Q} \omega(s) \cdot |a_v(s)|} \quad (2)$$

где Q – множество префиксов трасс журнала;

$\omega(s)$ – частота (счетчик) префикса s в журнале

$e_x(s)$ – множество действий, следующих за префиксом s трасс журнала

$a_v(s)$ – множество действий, непосредственно выполнимых из маркировки, в которую можно перевести модель, выполнив последовательность s .

Из определения следует, что всякое поведение, наблюдаемое в журнале, всегда будет представлено в модели процесса. Множество префиксов Q строится по предварительно вычисленным выравниваниям.

Оценка и прогнозирование параметров производительности

Как было сказано, помимо логической корректности, описываемой понятием бездефектности, бизнес-процесс должен характеризоваться определенными параметрами производительности [2]. Их наиболее точную оценку дает имитационное моделирование [16], однако оно ресурсоемко как по времени, так и по вычислительным ресурсам. Однако такие параметры как среднее время выполнения процесса и средние суммарные затраты, могут быть вычислены аналитически [17–19].

Формально время выполнения CT бизнес-процесса может быть рекурсивно вычислено на основании времен исполнения поддеревьев T_i следующим образом:

- Последовательное исполнение:

$$CT = \sum_{i=1}^n T_i \quad (3)$$

- Параллельное исполнение:

$$CT = \text{Max}(T_1, T_2, \dots, T_n) \quad (4)$$

- Условный выбор между поддеревьями с вероятностями p_1, p_2, \dots, p_n :

$$CT = \sum_{i=1}^n p_i T_i \quad (5)$$

Рассмотрим блок циклического исполнения на рис.4, где T_0 – длительность выполнения тела цикла, p_0 – вероятность завершения цикла, p_1, \dots, p_n – вероятности исполнения вариантов доработки, T_1, \dots, T_n – время исполнения вариантов доработки, T_{n+1} – время исполнения завершающего блока цикла.

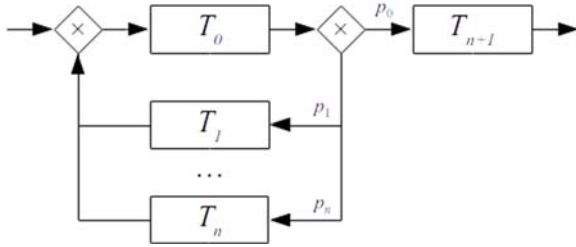


Рис. 4. Пример циклического процесса

Преобразуем схему, введя ещё один блок исключаяющего исполнения и обозначив вероятности переходов (рис. 5). Указанные на рисунке вероятности вариантов доработки результата получены на основе теоремы Байеса исходя из вероятности доработки $1 - p_0$.

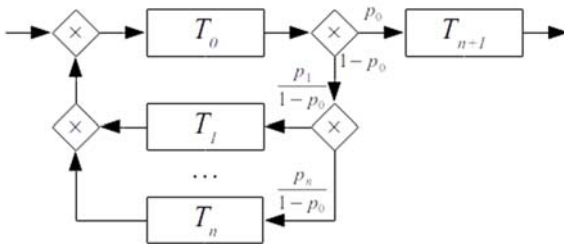


Рис. 5. Модифицированная схема циклического процесса

Полученный блок отображает случай, когда процессу требуется доработка. Заменяем блок исключаяющего исполнения элементарным блоком, считая вероятность перехода на данный блок: $p_r = 1 - p_0$, а время, по формуле (5):

$$T_r = \frac{1}{1-p_0} \sum_{i=1}^n T_i p_i.$$

Результат замены показан на рисунке 6.

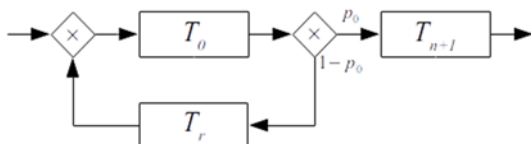


Рис. 6. Схема циклического процесса после замены

Узел окончания цикла и узел тела цикла выполняются в любом случае. Дополнительное время потребуется в случае доработок, т. е. получаем:

$$CT = T_0 + (T_r + T_0) \cdot (1 - p_0) + (T_r + T_0) \cdot (1 - p_0)^2 + \dots + T_{n+1} = (T_r + T_0) \cdot (1 + (1 - p_0) + (1 - p_0)^2 + \dots) - T_r + T_{n+1}$$

Применив формулу для суммы бесконечно убывающей геометрической прогрессии, получим итоговую оценку времени выполнения циклического процесса:

$$CT = \frac{T_r + T_0}{1 - (1 - p_0)} - T_r + T_{n+1} = \frac{T_0 + T_r(1 - p_0)}{p_0} + T_{n+1} = \frac{T_0 + \sum_{i=1}^n T_i p_i}{p_0} + T_{n+1} \quad (7)$$

Указанные соображения можно использовать для прогноза времени завершения с запущенным и частично исполненным процессом с текущим журналом выполнения L . Вычисление также производится рекурсивно от маркировки, соответствующей последней записи журнала. Для узлов последовательного, параллельного и исключаяющего поведения методика схожа с формулами (3), (4), (5). При вычислении времени доработки для циклов рассматривается два случая: если маркер в момент вычисления находится в теле цикла (количество входов в тело больше количества выходов), и если маркер находится в блоке доработки (количество входов равно количеству выходов).

Пусть n – количество входов в тело цикла,

$CT_r = \frac{\sum_{i=1}^n T_i p_i}{1 - p_0}$ – среднее время доработки. Результирующая формула для первого случая:

$$CT = (CT_r + T_0) \left(1 + \frac{(1 - p_0)^n}{p_0}\right) \quad (8)$$

Для случая, когда на момент вычисления маркер находится в теле доработки:

$$CT = (CT_r + T_0) \left(1 + \frac{(1 - p_0)^{n+1}}{p_0}\right) \quad (9)$$

Экспериментальные исследования

Описанные алгоритмы были реализованы в разработанной системе электронного документооборота совместно с машиной исполнения бизнес-процессов Onegine [7]. Согласно принятой методологии [19-21] эксперимент производился на созданных журналах событий процесса, содержащего порядка 20 действий. В журналы вносились случайные пропуски действий (от 1 до 6). После чего производился расчет предложенными в работе алгоритмами и эталонной реализацией на инструментариим pm4ру [22,23]. В каждой из шести групп производилось от 20 (для 1 удаления) до 10000 (для 6 удалений) экспериментов. Результаты представлены на на рисунке 7.

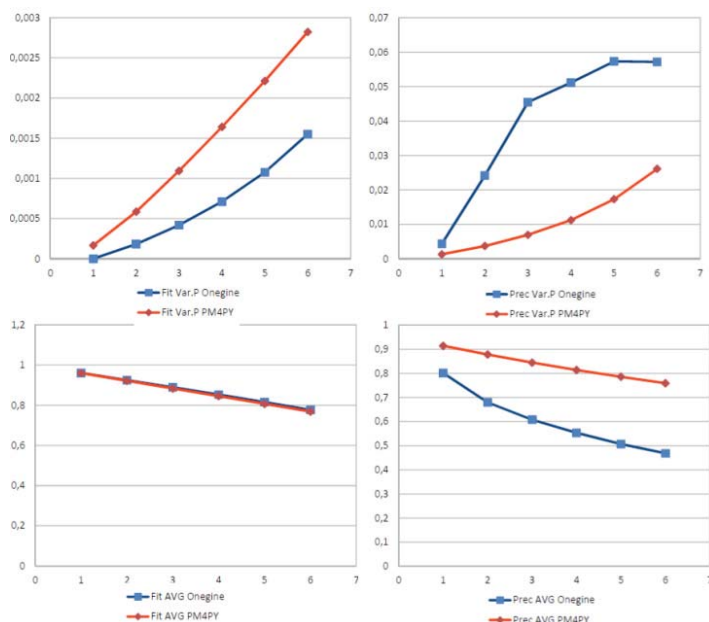


Рис. 7. Результаты экспериментального сравнения с pm4py

По метрике точности наблюдалось полное совпадение результатов (график 3) по математическому ожиданию значения для всех 6 типов экспериментов при несколько меньшей дисперсии с сохранением формы закономерности (график 1). Для метрики точности наблюдалось сохранение формы закономерности математического ожидания (график 4) при меньших значениях и иная форма дисперсии (график 2). Полученное противоречие может быть объяснено отсутствием, в отличие от метрики соответствия единого способа вычисления [24], что было подтверждено сравнительным изучением исходного кода pm4py и соответствующих плагинов ProM.

Заключение

В представленном прикладном исследовании были получены следующие выводы и результаты:

Реализованная схема и предложенные алгоритмы позволяют интегрировать средства описания, анализа и исполнения бизнес-процессов. Рассмотренные алгоритмы вычисления базовых метрик точности и соответствия показывают соизмеримые с известными инструментами результаты.

Использование платформы позволяет минимизировать использование сложных средств проверки корректности описания и повысить (до 2-3 раз) скорость внесения изменений в действующие бизнес-процессы.

Литература

1. W.M.P. van der Aalst. Process Mining: Discovery, Conformance and Enhancement of Business Processes. Berlin: Springer-Verlag, 2011. 352 p.
2. Васильев Н.В., Яшин А.И. Введение в анализ процессов. Учебное пособие. СПб.: Издательство СПбГЭТУ, 2020. 96 с.
3. Васильев Н.В., Яшин А.И. Введение в структурный синтез процессов автоматизации управления (Process Mining). Учебное пособие. СПб.: Издательство СПбГЭТУ, 2021. 88 с.

4. Васильев Н.В., Забродин О.В., Куликов Д.В. Метод Process mining в системе защищенного электронного документооборота // Научные технологии в космических исследованиях Земли. 2018. Т.10. №6. С. 38-50.
5. M. McGregor. The death of process mining? [Online]. Available: <https://www.bptrends.com/the-death-of-process-mining>, 2019.
6. W.M.P. van der Aalst. Process Mining: Bridging Not Only Data and Processes, but Also Industry and Academia (Blog Post Celonis). www.celonis.com, 2019.
7. Vasiliev N.V., Dorogov A.Y., Yashin A.I., Dovzhikov S.N. One-engine: an Engine for the Description, Analysis and Execution of Block-structured Business Processes // 2020 XXIII International Conference on Soft Computing and Measurements (SCM), St. Petersburg, Russia, 2020, pp. 165-168, doi: 10.1109/SCM50615.2020.9198819
8. Leemans S. J. J., Fahland D., W. M. P. van der Aalst. Discovering block-structured process models from incomplete event logs // Application and Theory of Petri Nets and Concurrency, G. Ciardo and E. Kindler, Eds. Cham: Springer International Publishing, 2014, pp. 91-110.
9. W.M.P. van der Aalst, J. Carmona, T. Chatain, and B.F. van Dongen. A Tour in Process Mining: From Practice to Algorithmic Challenges. In M. Koutny, L. Pomello, and L.M. Kristensen, editors // Transactions on Petri Nets and Other Models of Concurrency (ToPNoC 14), vol. 11970 of Lecture Notes in Computer Science, pp. 1-35. Springer-Verlag, Berlin, 2019.
10. Kalenkova A.A., Burattin A., M. de Leoni, W.M.P. van der Aalst, and A. Sperduti. Discovering High-Level BPMN Process Models From Event Data // Business Process Management Journal, 25(5):995-1019, 2019.
11. W.M.P. van der Aalst. Everything You Always Wanted to Know About Petri Nets, but Were Afraid to Ask. In T.T. Hildebrandt, B.F. van Dongen, M. Röglinger, and J. Mendling, editors // International Conference on Business Process Management (BPM 2019), vol. 11675 of Lecture Notes in Computer Science, pp. 3-9. Springer-Verlag, Berlin, 2019.
12. Adriansyah, Arya & Dongen, B.F. & Aalst, Wil. Conformance Checking Using Cost-Based Fitness Analysis. The Journal of Physical Chemistry. 2011, pp. 55-64. 10.1109/EDOC.2011.12.
13. Sander J.J. Leemans, Dirk Fahland, Wil M.P. van der Aalst. Scalable process discovery and conformance checking. Software & Systems Modeling 17(2): 599-631, 2018.
14. D. Schuster, S.J. van Zelst, and W.M.P. van der Aalst. Alignment Approximation for Process Trees // Computing Research Repository (CoRR) in arXiv, abs/2009.14094, 2020
15. Augusto, R. Conforti, A. Armas-Cervantes, M. Dumas and M. La Rosa. Measuring Fitness and Precision of Automatically Discovered Process Models: A Principled and Scalable Approach // IEEE Transactions on Knowledge and Data Engineering, 2021. doi: 10.1109/TKDE.2020.3003258.
16. Bedilia Estrada-Torres, Manuel Camargo, Marlon Dumas, Luciano García-Bañuelos, Ibrahim Mahdy, Maksym Yerokhin. Discovering business process simulation models in the presence of multitasking and availability constraints // Data & Knowledge Engineering, Vol. 134, 2021, 101897, ISSN 0169-023X, <https://doi.org/10.1016/j.datak.2021.101897>.
17. Dumas M., Rosa M.L., Mendling J., Reijers H.A. Fundamentals of Business Process Management (2-ed.). Springer Publishing Company, Incorporated, 2018.
18. Schuh G., Gutzlaff A., Schmitz S., W.M.P. van der Aalst. Data-Based Description of Process Performance in End-to-End Order Processing // CIRP Annals, 69(1):381-384, 2020.
19. C.Y. Li, S. van Zelst, and W.M.P. van der Aalst. A Generic Approach for Process Performance Analysis using Bipartite Graph Matching. In C. Di Francescomarino, R. Dijkman, and U. Zdun, editors



// Workshop on Business Process Intelligence (BPI 2019), BPM 2019 Workshop Proceedings, vol. 362 of Lecture Notes in Business Information Processing, pp. 199-211. Springer-Verlag, Berlin, 2019.

20. Augusto A. *et al.* Automated Discovery of Process Models from Event Logs: Review and Benchmark // IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 4, pp. 686-705, 1 April 2019, doi: 10.1109/TKDE.2018.2841877

21. Vasiliev N. V., Yashin A. I., Dovzhikov S. N. Integration of Business Process Definition, Analysis and Execution Tools Based on Block-structured Approach // 2021 10th Mediterranean Conference on Embedded Computing (MECO), 2021, pp. 1-6, doi: 10.1109/MECO52532.2021.9460283

22. Berti, Alessandro & van Zelst, Sebastiaan. Process Mining for Python (PM4Py): Bridging the Gap Between Process- and Data Science. 2019.

23. Berti A., S.J. van Zelst, and W.M.P. van der Aalst. PM4Py Web Services: Easy Development, Integration and Deployment of Process Mining Features in any Application Stack // Proceedings of the BPM Demo Track at BPM 2019, volume 2420 of CEUR Workshop Proceedings, pp. 174-183. CEUR-WS.org, 2019.

24. Augusto A. *et al.* Automated Discovery of Process Models from Event Logs: Review and Benchmark // IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 4, pp. 686-705, 1 April 2019, doi: 10.1109/TKDE.2018.2841877

INTEGRATION OF EXECUTION, PROCESS MINING AND PERFORMANCE ANALYSIS OF BUSINESS PROCESSES BASED ON A BLOCK APPROACH

NICKOLAY V. VASILIEV

Saint-Petersburg, Russia, gandvik1984@gmail.com

ALEXANDER I. YASHIN

Saint-Petersburg, Russia, a_yashin@inbox.ru

SERGEY N. DOVZHIKOV

Saint-Petersburg, Russia, sergei.dovzhikov@gmail.com

KEYWORDS: *process trees, verification, business process execution machines, Petri nets, business process performance analysis.*

ABSTRACT

Introduction: the core of modern enterprise management information systems are machines for the execution of business processes. The vast majority of them use the BPMN2 description language (for example, JBPM, Activiti, Camunda). Despite the adoption of the standard, the editors' support for BPMN is platform-oriented and has limited capabilities for transferring descriptions. To create and maintain BPMN descriptions up to date, a business analyst, in addition to editors, uses a whole set of auxiliary tools, including verification tools (for example, WoPeD) and performance analysis (Bizagi, BIMP). The attraction for identifying and updating business processes of the methodology of in-depth analysis of processes (ProcessMining), which uses its own set of tools (Disco, ProM), in which BPMN support is limited. As a result, the integration of analysis tools and tools for the execution of business processes remains insufficient, which significantly slows down the implementation and updating of enterprise management systems. **Purpose of the study:** based on the general log processing methodology used by Process Mining, show that a block-structured approach based on process trees can be a universal formalism at all stages (reconstruction, quality analysis, analysis of time and cost parameters). **Methods:** theory of Petri nets, discrete mathematics, mathematical logic, theory of algorithms. Results: the scheme of integrated analysis of busi-

ness processes is described. A tree dynamics model is proposed and implemented, which is the basis for the implementation of a business process execution machine. On the basis of the dynamics model, an algorithm for constructing the reachability graph of process tree markings is proposed. Conformity and accuracy assessment algorithms have been modified for their application with process trees. Within the framework of the methodology for the analytical assessment of the performance parameters of block business processes, a formula for estimating the completion time of a cyclic block with several revision options is obtained. An experimental study of the effectiveness of the developed solution was carried out by means of full-scale modeling of the integrated analysis scheme and the proposed methods and its comparison with industrial instruments (pm4py) according to the criteria of accuracy and compliance of the reconstruction. The experimental results obtained make it possible to speak about characteristics comparable with the known instruments. This reduces the need for external tools, making changes faster. **Practical significance:** the ideas presented in the work were implemented in the Onengine virtual machine developed by the authors and tested in a number of experimental design works. The results described in the work can serve as a basis for creating more effective enterprise management systems.

REFERENCES

1. W.M.P. van der Aalst (2011). *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Berlin: Springer-Verlag. 352 p.
2. N.V. Vasiliev, A.I. Yashin (2020). *Introduction to process analysis. Tutorial (in Russian)*. SPb.: Publishing house SPbETU. 96 p.
3. N.V. Vasiliev, A.I. Yashin (2021) to the structural synthesis of control automation processes (Process Mining). *Tutorial (In Russian)*. SPb.: Publishing house SPbETU. 88 p.
4. N.V. Vasiliev, O.V. Zabrodin, D.V. Kulikov (2018). Process mining method in a secure electronic document management system (in Russian). *Science-intensive technologies in space research of the Earth*. T.10. No. 6, pp. 38-50.
5. M. McGregor (2019). death of process mining? [Online]. Available: <https://www.bptrends.com/the-death-of-process-mining>.
6. W.M.P. van der Aalst (2019). *Process Mining: Bridging Not Only Data and Processes, but Also Industry and Academia (Blog Post Celonis)*. www.celonis.com.
7. N.V. Vasiliev, A.Y. Dorogov, A.I. Yashin and S.N. Dovzhikov, (2020). Onegine: an Engine for the Description, Analysis and Execution of Block-structured Business Processes. *2020 XXIII International Conference on Soft Computing and Measurements (SCM)*, St. Petersburg, Russia, pp. 165-168, doi: 10.1109/SCM50615.2020.9198819
8. S.J.J. Leemans, D. Fahland, and W.M.P. van der Aalst, (2014). Discovering block-structured process models from incomplete event logs. *Application and Theory of Petri Nets and Concurrency*, G. Ciardo and E. Kindler, Eds. Cham: Springer International Publishing, pp. 91-110.
9. W.M.P. van der Aalst, J. Carmona, T. Chatain, and B.F. van Dongen (2019). A Tour in Process Mining: From Practice to Algorithmic Challenges. In M. Koutny, L. Pomello, and L.M. Kristensen, editors, *Transactions on Petri Nets and Other Models of Concurrency (ToPNoC 14)*, volume 11970 of *Lecture Notes in Computer Science*, pp. 1-35. Springer-Verlag, Berlin.
10. A.A. Kalenkova, A. Burattin, M. de Leoni, W.M.P. van der Aalst, and A. Sperduti (2019). Discovering High-Level BPMN Process Models From Event Data. *Business Process Management Journal*, no.25(5), pp. 995-1019.
11. W.M.P. van der Aalst (2019). Everything You Always Wanted to Know About Petri Nets, but Were Afraid to Ask. In T.T. Hildebrandt, B.F. van Dongen, M. Reglinger, and J. Mendling, editors, *International Conference on Business Process Management (BPM 2019)*, vol. 11675 of *Lecture Notes in Computer Science*, pp. 3-9. Springer-Verlag, Berlin.
12. Adriansyah, Arya & Dongen, B.F. & Aalst, Wil. (2011). Conformance Checking Using Cost-Based Fitness Analysis. *The Journal of Physical Chemistry*, pp. 55-64. 10.1109/EDOC.2011.12.
13. Sander J.J. Leemans, Dirk Fahland, Wil M.P. van der Aalst. (2018). Scalable process discovery and conformance checking. *Software & Systems Modeling*, no. 17(2), pp. 599-631.
14. D. Schuster, S.J. van Zelst, and W.M.P. van der Aalst (2020). Alignment Approximation for Process Trees. *Computing Research Repository (CoRR)* in arXiv, abs/2009.14094.
15. Augusto, R. Conforti, A. Armas-Cervantes, M. Dumas and M. La Rosa (2021). Measuring Fitness and Precision of Automatically Discovered Process Models: A Principled and Scalable Approach. *IEEE Transactions on Knowledge and Data Engineering*, doi: 10.1109/TKDE.2020.3003258
16. Bedilia Estrada-Torres, Manuel Camargo, Marlon Dumas, Luciano Garcia-Baiuelos, Ibrahim Mahdy, Maksym Yerokhin (2021). Discovering business process simulation models in the presence of multitasking and availability constraints. *Data & Knowledge Engineering*. Vol. 134. P. 101897, ISSN 0169-023X, <https://doi.org/10.1016/j.datak.2021.101897>.
17. M. Dumas, M. L. Rosa, J. Mendling, and H. A. Reijers (2018). *Fundamentals of Business Process Management (2-ed.)*. Springer Publishing Company, Incorporated.
18. G. Schuh, A. Gutzlaff, S. Schmitz, and W.M.P. van der (2020). Data-Based Description of Process Performance in End-to-End Order Processing. *CIRP Annals*, no. 69(1), pp. 381-384.
19. C.Y. Li, S. van Zelst, and W.M.P. van der Aalst (2019). A Generic Approach for Process Performance Analysis using Bipartite Graph Matching. In C. Di Francescomarino, R. Dijkman, and U. Zdun, editors, *Workshop on Business Process Intelligence (BPI 2019), BPM 2019 Workshop Proceedings*, vol. 362 of *Lecture Notes in Business Information Processing*, pp. 199-211. Springer-Verlag, Berlin.
20. A. Augusto et al. (2019). "Discovery of Process Models from Event Logs: Review and Benchmark. *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 4, pp. 686-705, 1 April 2019, doi: 10.1109/TKDE.2018.2841877
21. N. V. Vasiliev, A. I. Yashin and S. N. Dovzhikov (2021). Integration of Business Process Definition, Analysis and Execution Tools Based on Block-structured Approach. *2021 10th Mediterranean Conference on Embedded Computing (MECO)*, pp. 1-6, doi: 10.1109/MECO52532.2021.9460283
22. Berti, Alessandro & van Zelst, Sebastiaan (2019). Process Mining for Python (PM4Py): Bridging the Gap Between Process- and Data Science.
23. A. Berti, S.J. van Zelst, and W.M.P. van der Aalst (2019). PM4Py Web Services: Easy Development, Integration and Deployment of Process Mining Features in any Application Stack. *Proceedings of the BPM Demo Track at BPM 2019*, vol. 2420 of *CEUR Workshop Proceedings*, pp. 174-183. CEUR-WS.org.
24. A. Augusto et al. (2019). Automated Discovery of Process Models from Event Logs: Review and Benchmark. *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 4, pp. 686-705, 1 April 2019, doi: 10.1109/TKDE.2018.2841877

INFORMATION ABOUT AUTHORS:

Vasiliev N.V., PhD., Head division of department Joint-Stock Company "Inteltech"

Yashin A.I., DrSc., engineer, CEO's counselor Joint-Stock Company "Inteltech"

Dovzhikov S.N., engineer, Joint-Stock Company "Inteltech"

For citation: Vasiliev N.V., Yashin A.I., Dovzhikov S.N. Integration of execution, process mining and performance analysis of business processes based on a block approach // H&ES Reserch. 2021. Vol. 13. No. No 6. P. 60-68. doi: 10.36724/2409-5419-2021-13-6-60-68 (In Rus)