КЛАССИФИКАЦИЯ АЛГОРИТМОВ ИЗВЛЕЧЕНИЯ ПРОЦЕССОВ ИЗ ЖУРНАЛОВ СОБЫТИЙ ИНФОРМАЦИОННЫХ СИСТЕМ

Бокарева Д. Г. [a], Земцов М. А. [b]
студент [a]
старший преподаватель кафедры систем автоматизации управления [b]
Вятский государственный университет, Киров, Российская Федерация [a, b]
E-mail: dariabokareva@gmail.com [a]; ma_zemtsov@vyatsu.ru [b]

Аннотация: *Извлечение и анализ процессов (Processmining) –современная область исследований, являющаяся связующим звеном между такими областями, как управление бизнес-процессами (BPM, Business Process Management) и интеллектуальный анализ данных (Data Mining).Использование методов Processmining даёт возможность получить объективную информацию о ходе процессов, так как использует фактические данные, извлекаемые из информационных систем (ИС). Первоочередной задачей данной технологии является построение моделей процессов на основеданных ИС. Для этого было разработано множество алгоритмов, различающихся типами выходных данных, скоростью работы, точностью выполнения и др. В статье рассмотрены основные алгоритмы, использующиеся для извлечения (восстановления) процессов из журналов событий информационных систем, и произведена их классификация.*

Ключевые слова: *Processmining, журнал событий, сети Петри, альфа-алгоритм, генетический алгоритм, теория регионов, индуктивный алгоритм.*

Введение. Извлечение процессов(интеллектуальный анализ процессов, Processmining) – общее название ряда методов и подходов, предназначенных для анализа и усовершенствования процессов в информационных системах или бизнес-процессов на основе изучения системных данных о выполненных операциях. Такие данные, как правило, хранятся в журналах событий (eventlogs) любых информационных систем.

Существует три основных раздела методов интеллектуального анализа процессов.

1. Обнаружение (Processdiscovery) – построение модели процесса на основании данных журнала событий информационной системы за некоторый промежуток времени.

2. Проверка соответствия (Conformancechecking) – сравнение существующей модели с журналом событий и анализ выявленных несоответствий.

3. Усовершенствование (Modelenhancement) – дополнение модели знаниями, полученными из журнала событий, и устранение несоответствий [1].

Для извлечения и анализа процессов было разработано множествоалгоритмов следующих видов: альфа-алгоритмы, вероятностные, эвристические алгоритмы, алгоритмы с использованием линейного программирования, основанные на теории регионов, индуктивные алгоритмы и др. Подробное описание принципов работы каждого алгоритма можно найти в [1, 3–10]. Результатом работы алгоритмов является модель процесса в виде сети Петри, сети потоков работ (Workflow-сети), нечеткой модели (Fuzzymodel), дерева процессов или др. Для работы с алгоритмами Processmining существует программная среда с открытым исходным кодом – ProM Framework.

Целью данной работы является обзор существующих алгоритмов восстановления процессов и выявление преимуществ и недостатков.

Результаты исследований, их обсуждение. Прежде чем приступить к классификации алгоритмов, следует выделить основные понятия, играющие важную роль в технологии Processmining, такие как сети Петри и Workflow-сети, а также рассмотреть метрики качества извлекаемых моделей.

Сети Петри (Petrinet) – математический аппарат для моделирования динамических дискретных систем. Сеть Петри представляет собой двудольный ориентированный мультиграф, состоящий из вершин двух типов — позиций и переходов, соединённых между собой дугами. Вершины одного типа не могут быть соединены непосредственно. В позициях могут размещаться метки (маркеры), способные перемещаться по сети [2]. Позиции в сети Петри обозначаются кругами, метки – закрашенными кругами внутри позиций, а переходы – прямоугольниками.

Workflow-сети (WF-сети) –сеть потоков работ, подкласс сетей Петри. Формализм WF-сетей введён основоположником Processmining Wil van der Aalst для моделирования потоков работ в Workflow-системах [3].

Сеть Петриявляется WF-сетью при выполнении следующих условий:

– существует только одна начальная позиция (*i*) и одна конечная позиция (*o*);

– каждый узел данной сети расположен на пути от *i* к *o*.

Для анализа процессов каждый узел WF-сети необходимо именовать. На рисунке 1 представлен пример WF-сети, где*i* – начальное событие (позиция),

содержащее в себе маркер (закрашенный круг), *o*–конечное событие, A, B, C, D, E, F-именованные узлы (переходы).

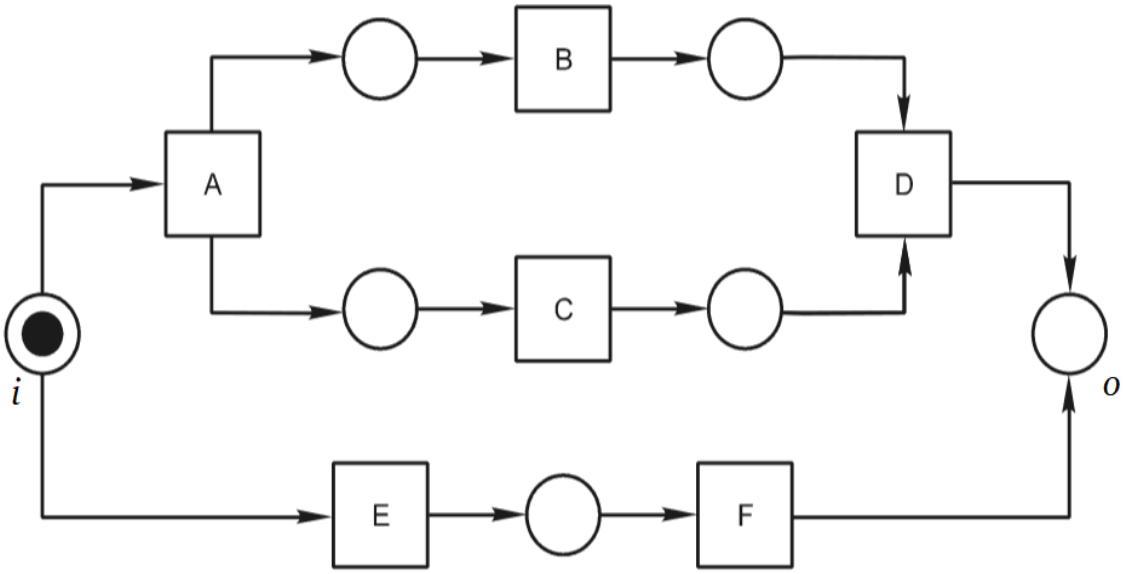

Рис. 1. Пример Workflow-сети

Для оценки качества восстановленной модели процесса обычно используются следующие метрики, подробно описанные в:

– «Пригодность» (fitness) – насколько хорошо модель описывает журнал. Чем больше экземпляров бизнес-процесса из журнала описывается моделью, тем выше мера пригодности;

– «Простота» (simplicity) – сложность модели бизнес-процесса приводит к ее бесполезности с точки зрения последующего анализа. Существует класс алгоритмов восстановления процессов, которые выдают хорошее значение метрики «простота», за счет снижения показателей «пригодности» и «точности»;

– «Точность» (precision) – строгое присутствие всех полученных вариантов выполнения процесса в журнале событий. Очень легко построить универсальную простую модель, которая может воспроизвести любую конечную последовательность событий указанных типов. У такой модели будет высокая «пригодность», но низкая «точность», так как будет очень много вариантов выполнения процесса, которыедопускаются моделью, но не присутствуют в журнале;

– «Обобщённость» (generalization) – метрика, по смыслу являющаяся противоположностью «точности». Можно построить такую модель, в которой каждая отдельная ветка будет предназначена для отражения каждого отдельного экземпляра бизнес-процесса из журнала. Подобная модель характеризуется высокой мерой «пригодности», но низкой мерой «обобщенности», так как не будет позволять ни малейшей модификации экземпляров процесса в журнале [11].

Алгоритмы извлечения процессов обычно заранее направлены на удовлетворение определённым метрикам. Следует отметить, что невозможно получить модель, имеющую высокую оценку по каждой метрике. Так как, к примеру, максимизация метрики «пригодности» снижает «простоту», а «обобщённость» противопоставляется «точности». Значение определённых метрик стоит принимать во внимание в зависимости от цели восстановления процесса.

Сети Петри являются универсальным и понятным представлением моделей процесса и используются в качестве конечного результата работы таких алгоритмов, как альфа-алгоритм, индуктивный алгоритм иалгоритмы с использованием линейного программирования, основанные на теории регионов (Region-basedalgorithms). Стоит отметить, что все исследуемые в данной работе алгоритмы так или иначе моделируют сети Петри в процессе исполнения. Алгоритмы, на выходе которых модель процесса представлена в отличном от сети Петри виде, моделирует последнюю на первом этапе исполнения. Далее, уже полученная сеть Петри преобразуется в модель иного вида (например, дерево процессов).

Альфа-алгоритм – наиболее распространённый алгоритм извлечения процессов, разработанный Ван-дер-Аальстом и др. [1, 3] и реализованный в среде ProMв виде плагина Alpha Miner.Алгоритм работает на основе бинарных отношений (следствия, причины, параллельности и несвязанности) в журнале событий и отличается простотой, по этой причине Alpha Minerне содержит настроек. Несмотря на то, что данный алгоритм способен обрабатывать большое количество специальных конструкций потока выполнения, он имеет ряд ограничений, касающихся исходного журнала событий. В первую очередь, журнал не должен

содержать «шум», это означает, что в журнале не должно быть пропущенных, некорректных и незавершённых событий. Шум может возникнуть в результате системных сбоев, неправильного ведения журнала событий и других факторов. Это немного усложняет работу с алгоритмом, так как предварительно журнал событий необходимо фильтровать. Кроме того, традиционный альфа-алгоритм не способен обрабатывать следующие специальные конструкции: «короткие циклы» (shortloops), «несвободный выбор» (non-free-choiceconstructs), «невидимые переходы» (invisible tasks), «задублированные переходы» (duplicate tasks).

С целью решения данных проблем, альфа-алгоритм был дополнен множеством расширений. Первое из них – α+-алгоритм, основанный на отношении следования. Он подсчитывает частоту отношения следования в логе событий, чтобы вывести оставшиеся отношения причины, параллельности и несвязанности. Главная идея работы алгоритма состоит в следующем заключении: если задача A чаще следует за задачей B, а задача B менее часто следует за A, то вероятность того, что A является причиной для B, выше. Таким образом, данный алгоритм способен работать с шумом, однако нелокальные конструкции «несвободный выбор» он выявить не может. Следующее расширение –Tsinghua-α-алгоритм,способный извлекать структурированные сети потоков работ (SWF-сети) с короткими циклами. Алгоритм использует пересекающиеся времена выполнения задач, чтобы различить параллелизм и короткие циклы α++-алгоритм, в свою очередь, извлекает сети Петри с локальными и нелокальными конструкциями «несвободный выбор». Расширения α#-алгоритм (β-алгоритм) и α*-алгоритмбазируются на методах работы α+-алгоритма. Отличительной особенностью β-алгоритма является использование им того факта, что задачи занимают время, а значит, параллелизм может быть явно обнаружен. В свою очередь, α*-алгоритмподдерживает работу с «задублированными переходами», однако в данных расширениях остаётся актуальной проблема поддержки «non-free-choice» конструкций.

Несмотря на то, что альфа-алгоритм является основным алгоритмом Process Mining, его нельзя назвать самым эффективным, поскольку ни одно его

расширение не решает проблем поддержки всех возможных конструкций и корректной работы с шумом. Следует отметить, что альфа-алгоритм направлен на максимизацию метрики простоты модели процесса.

Работа алгоритма, основанного на теории регионов [7], заключается в объединении минимальных групп состояний, предварительно организованных в ориентированный граф переходов из состояния в состояние, обладающих определёнными свойствами (например, все дуги у состояний из этой группы – входящие для каждого конкретного состояния), объединяются в регион – часть сети Петри, которая будет впоследствии построена. Таким образом, задачей теории регионов считается определение количества и мест позиций в сети Петри. В качестве расширения для данного алгоритмабыл предложен следующий метод: считать регионом только то множество переходов, которое не ограничивало бы описываемые процессы, если в регион добавить еще одно состояние, полученное специальным образом. Входящими дугами в состояние будут все входящие в регион дуги, выходящими дугами из состояния будут все выходящие из региона дуги. Существует также языковая теория регионов, где на вход передаётся не ориентированный граф переходов из состояния в состояние, а некоторый «язык», строящийся из алфавита, в качестве символов которого выступают события из журнала, а в качестве правил используются зависимости между событиями из журнала [12, с. 82]. Данные алгоритмы с использованием линейного программирования успешно обрабатывают большинство специальных конструкций сети Петри, не чувствительны к шумам и направлены на максимизацию метрики пригодности. К недостаткам алгоритма можно отнестипроблему «взрывного количества состояний» («stateexplosionproblem»), медленную обработку объёмных журналов событий, а также вероятность значительного ухудшения метрики точности при подаче на вход неполного лога относительно всех возможных ветвей процесса.Алгоритм реализован в ProMв виде плагина ILPMiner и содержит настройки, в частности, касающиеся производительности.

Индуктивный алгоритм – современный алгоритм, реализованный плагинами Inductive Miner и Inductivevisual Miner [14]. Индуктивный алгоритм рабо-

тает с использованием метода рекурсии: сначала он выбирает корневой оператор, соответствующий журналу событий, далее алгоритм разделяет действия в логе на непересекающиеся множества и при помощи этих множеств разбивает журнал событий на «поджурналы» (sublog), после чего рекурсивно восстанавливает данные поджурналы до тех пор, пока те не будут содержать только одно действие. Индуктивный алгоритм способен обрабатывать нечастые задачи, при чём он проверяет, насколько эти задачи влияют на модель процесса на каждом шаге рекурсии. Таким образом, он способен восстанавливать высококачественную и полную модель процесса, даже при содержании шума в журнале, а также отличается скоростью работы. Индуктивный алгоритм способен построить модель процесса не только в виде сети Петри, но и в виде дерева процессов (Process Tree).

Далее рассмотрим алгоритмы, конечным результатом которых является модель процесса в специфичном представлении. В первую очередь, это семейство эвристических алгоритмов [8]. Эвристический алгоритм реализован плагином Heuristic Mineri содержит довольно гибкие настройки, направленные на управление уровнем сложности извлекаемой модели.Работу алгоритма можно разделить на три основных шага: на первом шаге строится граф зависимостей между событиями в журнале (dependencygraph), второй шаг состоит из построения для каждой из вершин графа входных и выходных выражений и поиска зависимостей на большом расстоянии между событиями (расстояние по количеству событий), на последнем шаге на основе найденной информации строится модель процесса в виде эвристической сети (heuristicnet). Отличительной особенностью данного алгоритма является учет частотных характеристик в журнале событий. Полученная модель позволяет выделить в процессе наиболее частые и, наоборот, редкие маршруты, а предварительная настройка позволяет либо вывести подробную модель, либо исключить редкие события. Кроме того, эвристический алгоритм корректно работает с шумами в логе событий и поддерживает практически все основные конструкции. Из недостатков алгоритма – невозможность работы с «задублированными» событиями и необычный тип

модели процесса, однако последний недостаток компенсируется дополнительным плагином, поставляемым в пакете с Heuristic Miner и позволяющим конвертировать эвристические сети в сети Петри.

Генетические алгоритмы используют в своей работе методы эвристических алгоритмов [15]. А именно: на первом этапе с помощью методов эвристического алгоритма строится начальная популяция особей (т. е. возможных моделей процесса). Каждая особь в популяции представлена эвристической сетью и ей сопоставляется мера соответствия желаемому результату, которая характеризует ее качество. Мера соответствия – некоторая функция, оценивающая, насколько хорошо особь воспроизводит поведение в журнале событий. Популяции эволюционируют путем выбора наилучших особей и создания из них новых особей. Из работы алгоритма можно выделить его главное преимущество – качество конечной модели, но отсюда же вытекает и основной недостаток – низкая эффективность при работе с большим объемом данных. Тем не менее, алгоритм является достаточно гибким, устойчивым к шумам, а его расширение – генетический алгоритм с учётом повторений – поддерживает работу с «задублированными» событиями. Алгоритм реализован плагином Evolutionary Tree Miner, конечным результатом которого является дерево процессов (Process Tree).Следует отметить, что дерево процессовдостаточно просто преобразуется в сети Петри или BPMN-модель с помощью дополнительных плагинов, имеющихся в ProM.

Одним из самых молодых алгоритмов является нечёткий алгоритм (Fuzzyalgorithm) [9], схожий с эвристическим алгоритмом. Его работа направлена на решение проблемы большого числа возможных событий и неструктурированного поведения, соответственно его предпочтительно применять к сложным, неструктурированным журналам событий, а также с целью упрощения модели процесса. Суть работы плагина Fuzzy Miner, реализовавшего данный алгоритм, заключается в рассмотрении всех аспектов процесса и упрощении модели, с помощью интегрированного в него специального алгоритма, до требуемого уровня абстракции посредством удаления наименее важных собы-

тий или сокрытия их в кластеры. На выходе алгоритм восстанавливает модель процесса в нетипичном представлении, получившем название Fuzzymodel. Именно это считается основным недостатком алгоритма, так как данный тип модели невозможно преобразовать ни в один другой тип. Однако вместе с Fuzzymоделью алгоритм возвращает Fuzzyanimation, благодаря которой можно отследить динамику процесса.

Для работы с неструктурированным и сложным журналом также применяется вероятностный подход извлечения процессов. Данный подход реализован плагином Sequenceclustering [13], алгоритмом которого выступает цепь Маркова. В виде конечного результата представляются цепи Маркова (Trace Clustering)в заранее заданном при помощи настроек плагина количестве. Основная идея цепи Маркова состоит в использовании вероятностей последовательностей событий. В процессе работы алгоритма создаются таблицы вероятностей путем подсчета числа появлений одинаковых последовательностей в потоке событий. Затем последовательности, которые встречаются реже некоторого порога (порог устанавливается пользователем на этапе настроек), удаляются, а остальные используются для создания конечной модели. В результате мы можем получить одну и более моделей процесса в упрощенном виде. Кластеризация моделей происходит на основе выявления экземпляров процессов. Вероятностный алгоритм устойчив к шуму в журнале и поддерживает большинство конструкций, за исключением «дублируемых» задач.

Выводы. Таким образом, благодаря многообразию алгоритмов извлечения процессов, существует возможность выбрать наиболее оптимальный алгоритм восстановления модели процесса с учётом объёма и содержания журнала событий, а также целей исследования процессов.

Библиографический список

1. Van der Aalst W. M. P. Process mining: discovery, conformance and enhancement of business processes. Springer. 2011. P. 34–35.

2. Сети Петри. URL: https://ru.wikipedia.org/wiki/Сети_Петри. (дата обращения 30.03.2019).

3. Van der Aalst W.M.P., Weijters A. J. M. M., Maruster L. Workflow Mining: Discovering Process Models from Event Logs // IEEE Transactions on Knowledge and Data Engineering. 2004. Vol. 16(9). P. 1128–1142.

4. Lijie Wen, Jianmin Wang, Wil M. P. van der Aalst, Zhe Wang, Jiaguang Sun. A Novel Approach for Process Mining Based on Event Types // Tsinghua University and Eindhoven University of Technology, ISBN 90-386-2057-8/ISSN 1386–9213, May 2004. P. 163−190.

5. Van der Aalst W. M. P., Adriansyah A., Van Dongen B. F. Replaying history on process models for conformance checking and performance analysis // Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. Vol. 2(2). Wiley Online Library. 2012. P. 182–192.

6. Leemans S. J. J., Fahland D., Van der Aalst W. M. P. Discovering Block-Structured Process Models from Incomplete Event Logs // Tech. Rep. BPM-14-05. Eindhoven University of Technology. March 2014. P. 91–110.

7. Van der Werf J. M. E. M. et al. Process discovery using integer linear programming // Applications and Theory of Petri Nets. Springer Berlin Heidelberg. 2008. P. 368–387.

8. Weijters A., Van der Aalst W. M. P., De Medeiros A. K. A. Process mining with the heuristics miner-algorithm // TechnischeUniversiteit Eindhoven, Tech. Rep. WP. 2006. Vol. 166. P. 1–34.

9. Christian W. Gunther and Wil M.P. van der Aalst. Fuzzy Mining – Adaptive Process Simplification Based on Multi-perspective Metrics // Proceedings of the 5th International Conference on Business Process Management (BPM 2007) 24–28 September 2007. Brisbane, Australia. P. 328–343.

10. Maikel L. van Eck, Natalia Sidorova, and Wil M. P. van der Aalst. Composite State Machine Miner: Discovering and Exploring Multi-perspective Processes // In L. Azevedo, & C. Cabanillas (Eds.). Proceedings of the BPM Demo Track 2016. P. 73–77.

11. J. C. A. M. Buijs, B. F. van Dongen, and W. M. P. van der Aalst. On the Role of Fitness, Precision, Generalization and Simplicity in Process Discovery // In: OTM Conferences (1). vol. 7565, P. 305–322.

12. Анализ данных и процессов: учебное пособие // Барсегян А., Куприянов М., Холод И. и др. – 3-е изд.. СПБ: БХВ-Петербург. 2009. 512 стр.

13. Approaching Process Mining with Sequence Clustering: Experiments and Findings. Diogo Ferreira, Marielba Zacarias, Miguel Malheiros // Business Process Management: 5th International Conference, BPM 2007, Brisbane, Australia, September 24−28. 2007. P. 360–374.

14. Inductive visual Miner manual. Sander J. J. Leemans, ProM 6.7. June 7. 2017. P. 7–10.

15. Using Genetic Algorithms to Mine Process Models: Representation, Operators and Results. A. K. Alves de Medeiros, A. J. M. M. Weijters and W. M. P. van der Aalst // Eindhoven : Technische Universiteit Eindhoven. 2004. P. 38.

16. Методы когнитивного обучения: психолого-дидактический подход – Л. В. Ахметова ст. 48. URL: https://cyberleninka.ru/article/v/metody-kognitivnogo-obucheniya-psihologo-didakticheskiy-podhod (дата обращения 06.01.2020).

17. Когнитивный подход в обучении – Ковина Т. П. ст. 299−300. URL: http:// mospolytech.ru/science/aai77/scientific/article/s14/s14_70.pdf (дата обращения 07.01.2020).