

УДК 519.712.1

АЛГОРИТМ ФОРМИРОВАНИЯ ПРЕДИКАТНОЙ МОДЕЛИ ГИБКОГО МНОВАРИАНТНОГО ПРОЦЕССА НА ОСНОВЕ АНАЛИЗА ЕГО ЛОГ-ФАЙЛОВ

Рудометкина М.Н.

*ФГАОУ ВПО «Национальный исследовательский Томский политехнический университет»,
Томск, e-mail: mn.rud@inbox.ru*

В статье рассматривается реализация задачи построения иерархической модели гибкого дискретного процесса преобразования ресурсов. Предложен алгоритм формирования предикатной модели гибкого многовариантного процесса на основе анализа его лог-файлов. Данный алгоритм детализует метод построения иерархической предикатной модели гибкого процесса. Предикаты формализуют взаимосвязь между действиями процесса. Поэтому представление модели процесса в виде иерархии действий позволяет сопоставить модель с дополнительными данными из лог-файла, в частности с иерархией исполнителей процесса. По результатам сопоставления выполняется усовершенствование модели. Представление последовательности действий процесса в виде иерархии облегчает конфигурирование модели. Настройка выполняется путем удаления избыточных для конкретного применения ветвей дерева процесса. Разработанный алгоритм последовательно выполняет импорт файла лога, выделение множества базовых элементов процесса на основе анализа лога (последовательность, выбор, цикл, параллельное выполнение), построение иерархии действий процесса из полученного множества базовых элементов, а также экспорт модели по уровням иерархии.

Ключевые слова: предикат, граф, гибкий процесс, лог процесса, process mining

AN ALGORITHM FOR PREDICATE MODEL OF FLEXIBLE MULTIVARIANT PROCESS GENERATING BASED ON THE ANALYSIS OF THE ITS LOG FILES

Rudometkina M.N.

National Research Tomsk Polytechnic University, Tomsk, e-mail: mn.rud@inbox.ru

In the article the implementation of hierarchical model of flexible discrete process for resource conversion is examined. The algorithm for predicate model of flexible multivariant process on the base of its log-file analysis is proposed. This algorithm details the method of constructing a hierarchical predicate model of flexible process. Predicates formalize the relationship between the process actions. Therefore, the representation of the process model as a hierarchy of actions enables us to compare the model with additional data from the log file, in particular with the hierarchy of the process performers. As the result of comparison between the proposed algorithm and its analogs the improvement of the model is performed. The representation of actions sequence of the process as a hierarchy of models makes it easier to configure the model. Configuration is performed by removing the redundant tree branches for a particular application process. The proposed algorithm consistently performs next operations: importing of log-file, allocation of a set of basic process elements on the base of log-file analysis (sequence, choice, loop, concurrency), building the hierarchy of actions of the process from the resulting set of basic elements, as well as exporting model by the hierarchy levels.

Keywords: predicate, graph, flexible process, the log process, process mining

В данной работе рассматривается реализация задачи построения иерархической модели гибкого дискретного процесса преобразования ресурсов на основе анализа его лог-файлов. В качестве гибких процессов мы рассматриваем такие, структура которых может изменяться как на этапе конфигурирования, так и во время выполнения.

Актуальность данной задачи обосновывается следующими положениями. Во-первых, единая модель гибкого многовариантного процесса может быть адаптирована для близких предметных областей путем удаления избыточных последовательностей действий. Во-вторых, иерархическое представление модели гибкого процесса облегчает ее понимание, а также создает условия для ее усовершенствования на основе дополнительной информации из лога. В-третьих, построение моделей выполня-

ющихся процессов обработки ресурсов на основе их логов позволяет проверить соответствие модели реально работающему процессу и, при необходимости, устранить расхождения между ними.

Анализ работ области построения моделей гибких процессов

Проведенный анализ работ [1, 4, 5, 6] в области построения моделей гибких процессов как традиционными методами, так и средствами process mining показал, что полная модель такого процесса может быть получена:

- путем реализации традиционного полного цикла разработки модели гибкого многовариантного процесса;

- на основе слияния моделей нескольких «жестких» процессов, которые реализуют идентичную функциональность, но различным способом;

– путем поэтапного дополнения исходной модели процесса новыми возможностями;
 – на основе анализа логов, содержащих информацию о нескольких вариантах реализации процесса [9].

Общий недостаток рассмотренных подходов к построению моделей процессов состоит в том, что они базируются на предположении о его «двумерном» характере.

Автором предложен метод построения иерархической предикатной модели гибкого процесса [2], который при условии полноты лога обеспечивает построение полного адаптируемого описания процесса в форме иерархии предикатов, связывающих его базовые структурные элементы. Изложенное свидетельствует об актуальности реализации метода в виде соответствующего программного обеспечения и, следовательно, о необходимости разработки алгоритма построения иерархической модели гибкого процесса на основе анализа лог-файла.

Формулировка задачи

Задача построения модели гибкого процесса требует учета базовых элементов лог-файла, фиксирующих поведение процесса, базовых элементов процесса, важности многоуровневого представления детализации модели.

Исходными данными для разработки алгоритма подхода являются логи событий, отражающие последовательность выполнения действий при реализации процесса. На основе слияния и анализа логов выполняется построение полной многовариантной модели процесса [4].

Результирующая иерархическая модель [3] должна состоять из уровней, соответствующих иерархии исполнителей и отражающих взаимосвязь действий процесса на основе его базовых элементов.

Исходные данные: основные элементы лог-файлов

Исходными данными для разработанного алгоритма построения предикатной модели являются лог-файлы. Такие файлы создаются информационными системами и фиксируют их деятельность, происходящие в таких системах процессы. В указанных файлах с метками времени фиксируются выполняемые задачи, изменения в данных, изменения статуса системы и ее отдельных компонентов и т.д.

Файлы логов обычно содержат следующую информацию о выполнении процессов (доступа к информации, бухгалтерского учета и т.п.):

– код ситуации; каждая ситуация соответствует однократному выполнению про-

цесса; последовательность действий с одним кодом ситуации представляет собой «след» или траекторию однократного выполнения процесса;

– код события; событие отражает факт выполнения одного из действий процесса;

– метка времени; данная метка фиксирует факт выполнения одного из действий процесса, для которого составлен лог;

– наименование действия процесса;

– ресурс, который используется для выполнения действия процесса.

При построении иерархической модели процесса в соответствии с разработанным алгоритмом часть информации является избыточной и может не учитываться. При дальнейшем применении данного подхода, а также программного обеспечения возможно использование «избыточных» данных при построении иных аспектов модели процесса – организационного, информационного, управляющего, продуктового.

Лог может быть представлен в текстовом формате, в виде таблицы, а также в xml – подобном формате OpenXES. В этом случае лог хранится в файле с расширением «xes».

Основные теги, обеспечивающие структуризацию лога: trace – траектория выполнения (следа) процесса; event – событие, отражающее выполнение действия процесса.

Пример построения иерархической предикатной модели гибкого процесса

Рассмотрим процесс анализа лога. На первом шаге известны начальное и конечное состояние процесса, рис. 1.



Рис. 1. Начальное и конечное состояние процесса

Начинаем анализ с первого состояния процесса. Возможны два варианта продолжения процесса – развилка или последовательная цепочка действий. Развилка может быть пяти различных типов. После состояния x_1 в логе идет развилка в состояния x_2 и x_3 со свойством эксклюзивного выбора. Обозначим действия из x_1 в x_2 и из x_1 в x_3 соответственно бинарными предикатами $R_1(x_1, x_2)$ и $R_2(x_1, x_3)$, а развилку – предикатной операцией

$$S_1 = R_1(x_1, x_2) \oplus R_2(x_1, x_3).$$

Непроанализированные части лога обозначим буквами A и B , (рис. 2, а).

Продолжаем анализ процесса. Из состояния x_2 обнаруживается развилка в состояния x_4 и x_5 со свойством выбора (логическая операция OR). Из состояния x_3 продолжается последовательная цепочка к действию x_6 .

Обозначим действия из x_2 в x_4 , из x_2 в x_5 и из x_3 в x_6 соответственно бинарными предикатами

$R_3(x_2, x_4)$, $R_4(x_2, x_5)$ и $R_5(x_3, x_6)$,
а развилку – предикатной операцией
 $S_2 = R_1(x_1, x_2) \vee R_2(x_1, x_3)$.

Непроанализированные части процесса обозначим буквами A, B, C (рис. 2, б).

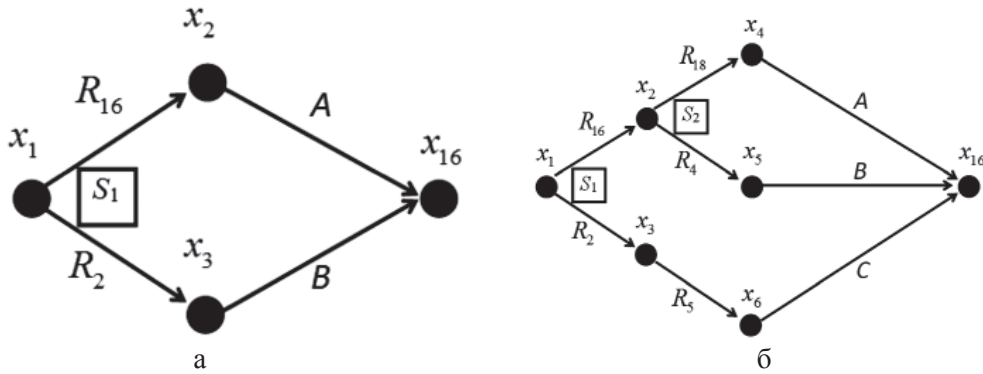


Рис. 2. Непроанализированные части

Проанализируем части процесса A, B и C (рис. 2, б). Из состояния x_5 обнаруживается развилка в состояния x_7 и x_8 со свойством выбора (логическая операция OR). Из состояний x_4 и x_6 продолжают последовательные цепочки к действиям x_9 и x_{11} соответственно.

Обозначим действия из x_4 в x_9 , из x_5 в x_7 , из x_5 в x_8 и из x_6 в x_{11} соответственно би-

нарными предикатами $R_6(x_4, x_9)$, $R_7(x_5, x_7)$, $R_8(x_5, x_8)$ и $R_9(x_6, x_{11})$ а развилку – предикатной операцией

$$S_3 = R_7(x_5, x_7) \vee R_8(x_5, x_8).$$

Непроанализированные части процесса обозначим буквами A, B, C, D (рис. 3, а).

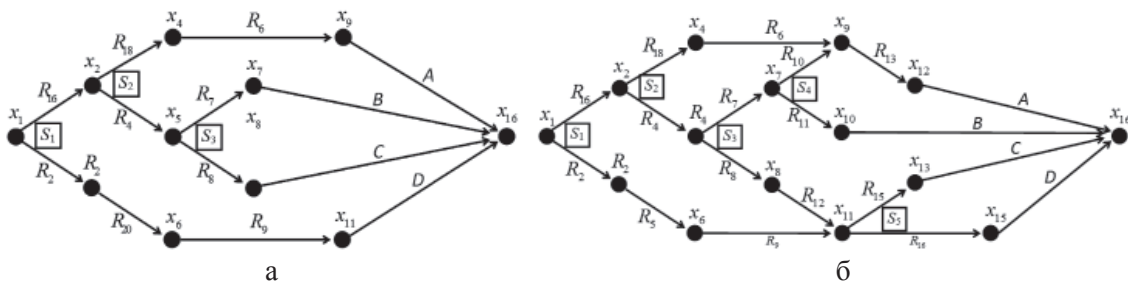


Рис. 3. Непроанализированные части процесса

Проанализируем части процесса A, B, C, D (рис. 3, а). Из состояния x_7 обнаруживается развилка в состояния x_9 и x_{10} со свойством параллельного выполнения (логическая операция AND). Из состояния x_{11} обнаруживается развилка в состояния x_{13} и x_{15} со свойством цикла. Из состояния x_9 продолжается последовательная цепочка к действию x_{12} .

Обозначим действия из x_7 в x_9 , из x_7 в x_{10} , из x_8 в x_{11} , из x_{11} в x_{13} и из x_{11} в x_{15} соответственно бинарными предикатами $R_{10}(x_7, x_9)$, $R_{11}(x_7, x_{10})$, $R_{12}(x_8, x_{11})$, $R_{15}(x_{11}, x_{13})$ и $R_{16}(x_{11}, x_{15})$. Развилку из состояния x_7 обозначим предикатной операцией

$$S_4 = R_{10}(x_7, x_9) \wedge R_{11}(x_7, x_{10}),$$

а развилку из состояния x_{11} обозначим

$$S_5 = \Omega(R_{15}(x_{11}, x_{13}), R_{16}(x_{11}, x_{15}), 3).$$

Непроанализированные части процесса обозначим буквами A, B, C, D (рис. 3, б).

Проанализируем части процесса A, B, C, D (рис. 3, б). Из состояний x_{12} , x_{10} , x_{13} и x_{15} продолжают последовательные цепочки к действиям x_{14} , x_{14} , x_{15} и x_{16} соответственно. Обозначим действия из x_{12} в x_{14} , из x_{10} в x_{14} , из x_{13} в x_{15} и из x_{15} в x_{16} соответственно бинарными предикатами $R_{17}(x_{12}, x_{14})$, $R_{14}(x_{10}, x_{14})$, $R_{18}(x_{13}, x_{15})$ и $R_{20}(x_{15}, x_{16})$.

Непроанализированную часть процесса обозначим буквой A (рис. 4).

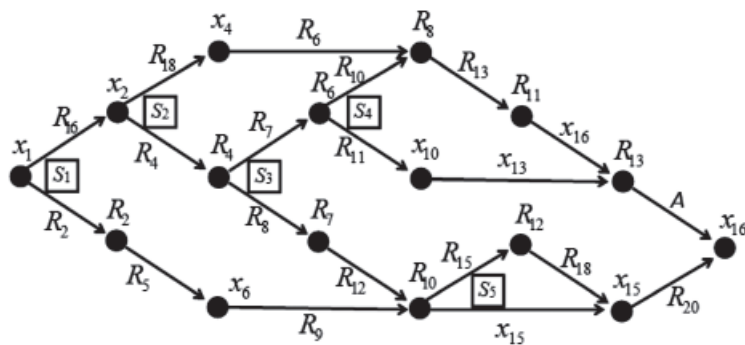


Рис. 4. Непроанализированная часть процесса A

Проанализируем части процесса A (рис. 4). Из состояния x_{14} продолжается последовательная цепочка к действию конечному x_{16} .

Обозначим действие из x_{14} в x_{16} бинарным предикатом $R_{19}(x_{14}, x_{16})$. На этом анализ процесса завершён, получаем граф логической сети.

Алгоритм формирования предикатной модели

Описание основных функций, реализация которых необходима для построения предикатной модели процесса, приведено в таблице.

Описание функций

Функции	Полное описание
Импорт файла лога	Данная функция выполняет импорт файла лога из текстового формата (или xml-подобного формата), его преобразование в единый внутренний формат и сохранение полученного стандартизованного лога. Полученный лог используется в дальнейших шагах по построению модели гибкого процесса
Формирование набора базовых элементов процесса	Данные функции предназначены для выделения базовых элементов процесса на основе сравнительного анализа ситуаций (следов процесса). В разработанном алгоритме используются следующие базовые элементы процесса [8, 9]: – последовательное выполнение действий процесса; – одновременное (параллельное) выполнение действий; – циклическое выполнение действий; – выбор последующих действий
Построение иерархической модели процесса	Данная функция обеспечивает собственно построение модели в виде иерархии. Уровни иерархии связаны между собой на основе приведенных выше базовых элементов процесса. Выделение уровней иерархии позволяет структурировать процесс путем выделения из него ряда подпроцессов
Экспорт модели	Формирование структурированного лога, отражающего различные уровни модели (по запросу). Полученные данные могут быть использованы в существующих внешних программах визуализации и анализа модели процесса на основе анализа логов

Диаграмма вариантов использования, построенная на основе рассмотренного перечня основных функций построения предикатной модели процесса, приведена на рис. 5.

Предлагаемый в работе алгоритм предназначен для построения модели гибкого процесса, в основе которой лежит дерево процесса. Использование иерархической модели позволяет выделить отдельные ветви, основанные на элементе выбора, и та-

ким образом обеспечить возможность отсеечения различных вариантов процесса.

Разработанный алгоритм в общем виде приведен на рис. 6. Его отличительная особенность – рекурсивное выполнение этапов построения иерархии процесса.

Отметим, что возможность внесения в модель дополнительных ветвей зависит от наличия различных вариаций лога. Отражение в модели нового лога означает возможное формирование новых ветвей процессного дерева.

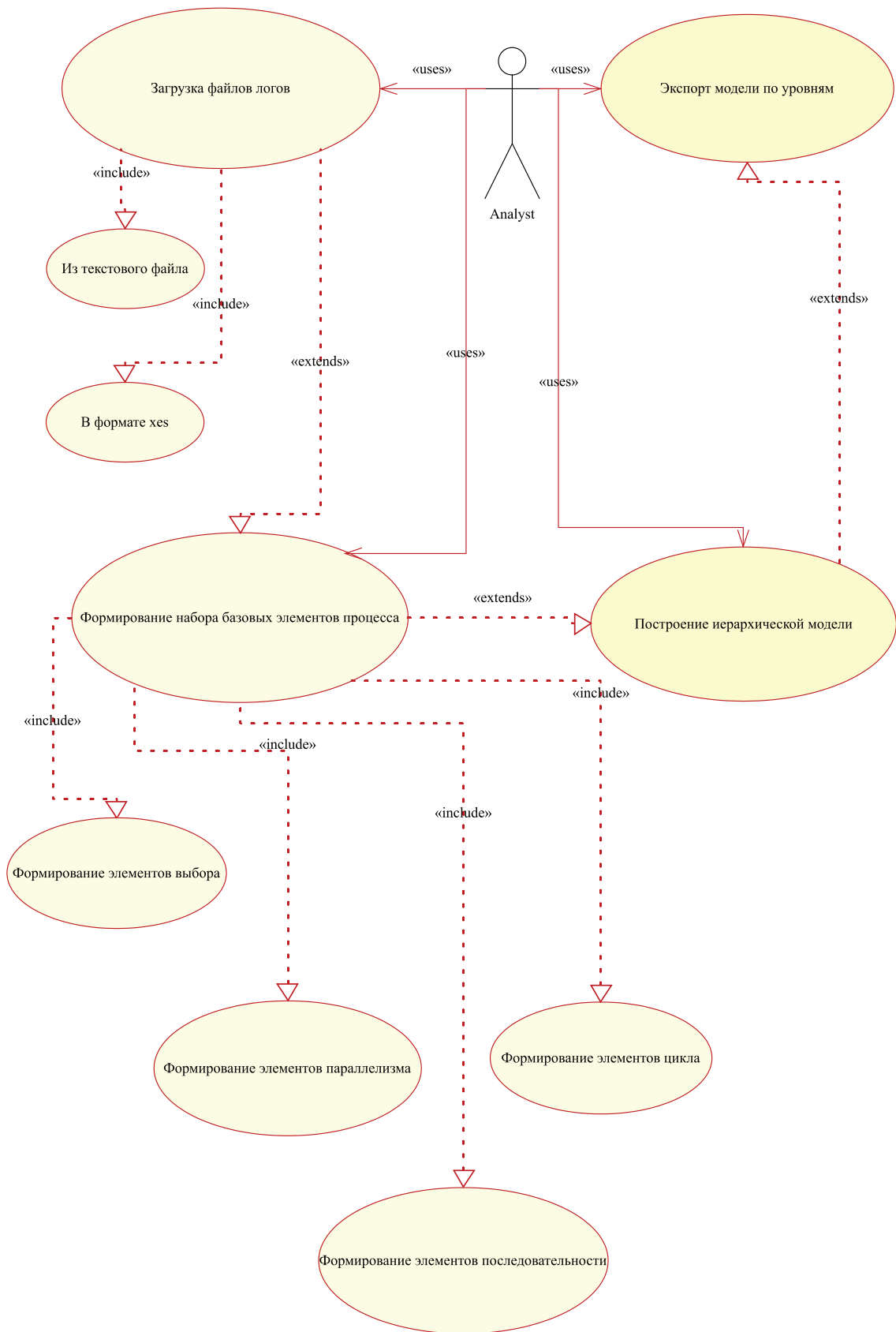


Рис. 5. Диаграмма вариантов использования

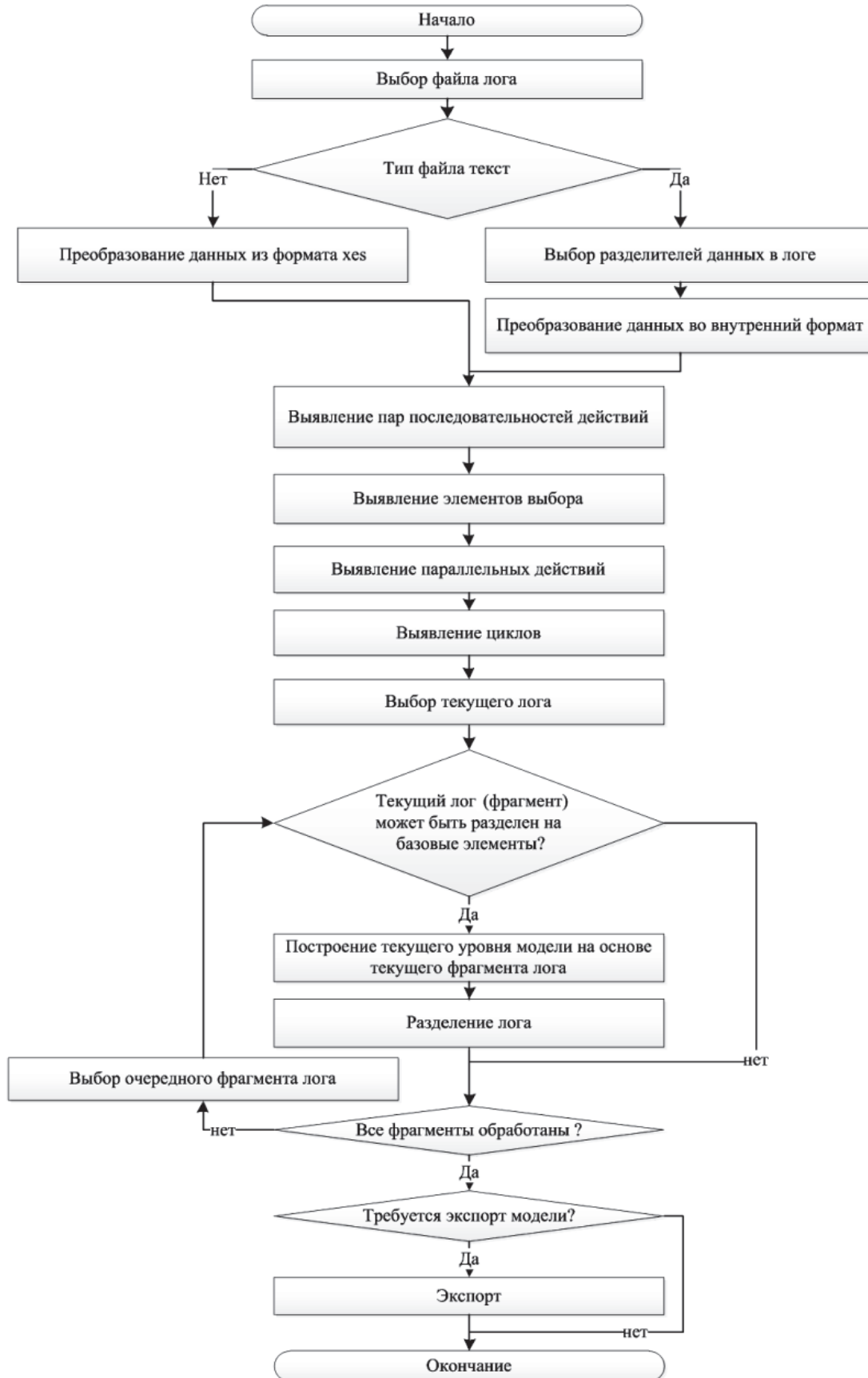


Рис. 6. Алгоритм формирования модели процесса

Заключение

В статье предложен алгоритм построения иерархической модели гибкого процесса на основе анализа логов, который реализует разработанный автором метод построения иерархической предикатной модели. Алгоритм предназначен для построения полного адаптируемого описания процесса в форме иерархии базовых элементов. Алгоритм учитывает наличие дополнительной информации при построении иерархии. Область практического применения алгоритма – интеллектуальный анализ процессов.

Работа выполнялась по тематике государственной НИР № 4.1316.2014 в рамках государственного задания «Наука».

Список литературы

1. Репин В., Елифиров В. Процессный подход к управлению. Моделирование бизнес-процессов. – М.: Стандарты и качество, 2008. – 196 с.
2. Рудометкина М.Н. Метод построения иерархической предикатной модели процесса // Фундаментальные исследования. – 2014. – № 9, Ч. 12.
3. Рудометкина М.Н. Предикатная модель гибкого процесса / Международный журнал прикладных и фундаментальных исследований. – 2014. – № 10, Ч. 2.
4. Gottschalk F., van der Aalst W.M.P., Jansen-Vullers M.H. Merging Event-driven Process Chains. OTM 2008, Part I, CoopIS 2008, volume 5331 of Lecture Notes in Computer Science. – Berlin Heidelberg, 2008. Springer Verlag. – P. 418–426.
5. La Rosa M., Dumas M., Uba R., Dijkman R. Business Process Model Merging: An Approach to Business Process Consolidation. ACM Transactions on Software Engineering and Methodology. – 2012. – № 22(2).
6. Li C., Reichert M., Wombacher A. The Minadept Clustering Approach for Discovering Reference Process Models Out of Process Variants // International Journal of Cooperative Information Systems. – 2010. – № 19(3-4). – P. 159–203.
7. Buijs J.C.A.M., van Dongen B.F., van der Aalst W.M.P. On the Role of Fitness, Precision, Generalization and Simplicity in Process Discovery. In Proceedings of Coop IS, LNCS. – Springer, 2012.
8. Rosemann M., W.M.P. van der Aalst. A Configurable Reference Modeling Language // Information Systems. – 2007. – № 32(1). – P. 1–23.
9. Rudometkina M.N., Spitsyn V.G. Detection of processing model basic elements in intellectual analysis of flexible

processes // IFOST, The 9th International Forum on Strategic Technology 2014.

References

1. Repin V., Elifirov V. Processnyj podhod k upravleniju. Modelirovanie biznes-processov. M.: Standarty i kachestvo, 2008. 196 p.
2. Pudometkina M.N. Metod postroenija ierarhicheskoj predikatnoj modeli processa Fundamental'nye issledovaniya no. 9, chast' 12, Rossijskoj Akademii Estestvoznaniya, Moskva, 2014.
3. Pudometkina M.N. Predikatnaja model' gibkogo processa Mezhdunarodnyj zhurnal prikladnyh i fundamental'nyh issledovaniy no. 10, chast' 2, Rossijskoj Akademii Estestvoznaniya, Moskva, 2014
4. Gottschalk F., van der Aalst W.M.P., Jansen-Vullers M.H. Merging Event-driven Process Chains. OTM 2008, Part I, CoopIS 2008, volume 5331 of Lecture Notes in Computer Science, pp. 418–426, Berlin Heidelberg, 2008. Springer Verlag.
5. La Rosa M., Dumas M., Uba R., Dijkman R. Business Process Model Merging: An Approach to Business Process Consolidation. ACM Transactions on Software Engineering and Methodology, no. 22(2), 2012.
6. Li C., Reichert M., Wombacher A. The MINADEPT Clustering Approach for Discovering Reference Process Models Out of Process Variants. International Journal of Cooperative Information Systems, no. 19(3–4):159–203, 2010.
7. Buijs J.C.A.M., van Dongen B.F., van der Aalst W.M.P. On the Role of Fitness, Precision, Generalization and Simplicity in Process Discovery. In Proceedings of Coop IS, LNCS. Springer, 2012.
8. Rosemann M. van der Aalst W.M.P. A Configurable Reference Modeling Language. Information Systems, no. 32(1):1–23, 2007.
9. Rudometkina M.N., Spitsyn V.G. Detection of processing model basic elements in intellectual analysis of flexible processes // IFOST, The 9th International Forum on Strategic Technology 2014.

Рецензенты:

Тузовский А.Ф., д.т.н., профессор, ФГАОУ ВПО «Национальный исследовательский Томский политехнический университет» Министерства образования и науки РФ, г. Томск;

Замятин А.В., д.т.н., профессор, ФГАОУ ВПО «Национальный исследовательский Томский государственный университет» Министерства образования и науки РФ, г. Томск.

Работа поступила в редакцию 01.10.2014.