

УДК 519.712.1

Использование настраиваемых базовых элементов при построении модели гибкого процесса*

М.Н. РУДОМЕТКИНА

634050, РФ, г. Томск, пр. Ленина, 30, Национальный исследовательский Томский политехнический университет, аспирант. E-mail: mn.rud@inbox.ru

В работе излагается метод построения модели гибкого процесса путем использования методов process mining с применением алгебры конечных предикатов и логических сетей. Интеллектуальный анализ процессов предназначен для построения формально-логических моделей гибких процессов. В качестве исходных данных используются логи событий, которые содержат сведения о последовательности выполненных действий в информационной системе. Алгебра конечных предикатов предназначена для формализации процессов логической природы и представляет собой дискретный аппарат для описания произвольных конечных отношений. Решение уравнений алгебры конечных предикатов осуществляется посредством логической сети, для чего уравнения алгебры преобразуются в систему бинарных предикатов.

Модель гибкого процесса объединяет набор возможных вариантов его выполнения. Конкретный вариант может быть выбран либо при начальном конфигурировании процессов в информационной системе, либо адаптирован при изменениях внешней среды во время выполнения процесса.

Предлагаемый подход заключается в последовательном дополнении модели различными вариантами реализации процесса на этапе анализа логов. Модель дополняется различными вариантами реализации гибкого процесса. Возможные варианты связей между действиями процесса представлены с помощью его базовых элементов. Выделение отличающихся элементов производится с помощью логических сетей.

Гибкость модели процесса достигается за счет применения операторов адаптации ее базовых элементов. Поскольку модель гибкого процесса содержит все возможные варианты его реализации, то адаптация обычно заключается в устранении избыточных возможностей. В качестве базовых элементов в модели используется последовательное, параллельное, циклическое выполнение, а также выбор последующих действий процесса.

Операторы обеспечивают статическую и динамическую адаптацию модели. Статическая адаптация выполняется на этапе конфигурирования модели в информационной системе и учитывает особенности предметной области. Динамическая адаптация осуществляется при реализации модели процесса в случае изменений условий внешнего окружения процесса.

* Статья получена 28 августа 2015 г.
Работа поддержана Грантом РФФИ № 12-08-00296.

Ключевые слова: интеллектуальный анализ, логические сети, гибкий процесс, логи событий, алгоритмы, модели, отношения, операторы адаптации процесса, статическая адаптация, динамическая адаптация

DOI: 10.17212/1814-1196-2015-1-107-120

ВВЕДЕНИЕ

Интеллектуальный анализ процессов (process mining) предназначен для построения моделей информационных процессов преобразования ресурсов [1, 7–9, 11–23]. Исходными данными для анализа являются записи о выполнении процессов, представленные в виде файлов – логов событий. Такие логи содержат сведения о последовательности произошедших событий в некоторой информационной системе с обязательными метками времени. Логи событий могут фиксировать выполнение различных составляющих (активностей) технологических процессов, равно как и бизнес-процессов или поведения пользователей в социальных сетях. Методы process mining базируются на интеллектуальном анализе данных и позволяют строить графовые модели для типовых процессов с жесткой структурой. В то же время сегодня серьезное внимание в данной области уделяется гибким процессам, структура которых может изменяться как на этапе конфигурирования, так и во время выполнения.

Структура процессов может изменяться вследствие влияния следующих основных факторов:

- скрытых неформализованных знаний о выполнении процесса, влияющих на последовательность его активностей;
- опыта исполнителей процесса, приводящего к изменению внутренней структуры активностей и, следовательно, изменяющего их результат;
- порядка взаимодействия (часто неформального) исполнителей – людей либо организаций при выполнении процесса;
- территориальной распределенности процесса, влияющей на порядок взаимодействия между его составляющими.

Сфера применения моделей таких гибких процессов достаточно широка – от научных исследований до современных технологий программирования (например, SCRUM) и гибких роботизированных систем. Например, алгебраический метод синтеза регуляторов пониженного порядка для линейных САУ, развитый А.В. Чехонадских и его соавторами [2–4], предполагает выполнение нескольких подобных операций при поиске оптимальных регуляторов различной структуры для разнообразных объектов. При попытке автоматизации процедуры такого синтеза в общем случае возникнет достаточно широкая сеть подпроцессов, не предусматривающая их жесткого упорядочения. Ее элементы будут включаться в процесс или исключаться из него в зависимости от специфики каждого конкретного задания.

В этой связи перспективной задачей представляется разработка алгоритмов process mining, которые обеспечивают построение моделей гибких процессов, отражающих различные варианты их выполнения. Модель гибкого процесса в дальнейшем может быть адаптирована к конкретным условиям выполнения в статическом и динамическом режимах.

Исходными данными для рассматриваемой задачи является набор логов событий, который содержит несколько вариантов реализации таких процессов.

Полученная модель процесса должна интегрировать несколько вариантов выполнения одного процесса. Граф процесса формализует последовательность его действий. Вершины графа соответствуют действиям процесса, а дуги задают порядок действий. Различие между вариантами достигается за счет использования дополнительных знаний о необходимости пропуска отдельных действий (или последовательностей таких действий), а также невыполнения отдельных ветвей в графе процесса. Таким образом, модель гибкого процесса содержит как общие элементы для всех вариантов реализации, так и знания об отличиях между этими вариантами. Все это облегчает выделение частных процессов на основе полученной модели гибкого процесса, а также повторное использование и адаптацию таких моделей.

1. АНАЛИЗ ИССЛЕДОВАНИЙ В ОБЛАСТИ ПОСТРОЕНИЯ МОДЕЛЕЙ ГИБКИХ ПРОЦЕССОВ

В последние годы активно проводятся исследования в области построения конфигурируемых моделей гибких процессов на основе использования различных источников информации (в том числе логов событий (event logs)).

Полная модель гибкого процесса может быть получена с помощью следующих подходов:

- выполнение традиционного полного цикла разработки модели гибкого многовариантного процесса;
- слияние моделей нескольких «жестких» процессов, которые реализуют идентичную функциональность, но различным способом;
- поэтапное дополнение исходной модели процесса новыми возможностями, например, при нахождении неявных знаний (используя опыт сотрудников), их описании и дальнейшем включении в модель;
- получение модели с помощью анализа логов, содержащих информацию о нескольких вариантах реализации процесса.

Первый подход основан на классических подходах к проектированию бизнес-процессов [5] и отличается дополнительными затратами на создание различных вариантов реализации процесса.

При втором и третьем подходах могут быть использованы алгоритмы, представленные в работах [11–23]. Общая идея этой группы алгоритмов состоит в том, что первоначально формируется базовая модель процесса, которая задает его «типовое» поведение (аналог традиционного «жестко заданного» процесса). Затем полученная модель дополняется возможными вариантами реализации.

Четвертый подход заключается в предварительном механическом объединении логов для нескольких вариантов реализации процесса и построении модели методами process mining. В данном случае преимущество имеют известные методы, основанные на эволюционном программировании [4].

В работах Gottschalk, W.M.P. van der Aalst предложен подход к построению модели гибкого процесса путем слияния пар Event-driven Process Chains

[11–13]. Это позволяет отразить в одной модели два варианта поведения процесса. При объединении моделей выполняется слияние функций (активностей процесса) путем объединения вершин, а также объединения либо разделения дуг (the split and join types of the arcs) соответствующих графов.

В работах La Rosa, M. Dumas, R. Dijkman рассматривается построение конфигурируемой модели процесса путем попарного слияния нескольких моделей [14–19]. При слиянии используются правила упрощения модели (Reduction Rules), что позволяет получить максимально компактную итоговую модель (as compact as possible). Достоинство данного алгоритма состоит в том, что можно итеративно объединить несколько моделей, задав в итоговой конфигурируемой модели все возможные варианты поведения процесса.

С. Li, M. Reichert и A. Wombacher предложен подход (a cluster-based approach) к формированию модели процесса с различными вариантами выполнения [20–23]. Алгоритм построения модели имеет полиномиальную сложность (has polynomial complexity), что позволяет сливать сложные модели при решении практических задач (real-world problems).

В отличие от предлагаемого в данной работе подхода, в работах [11–23] рассматривается только слияние существующих моделей. Дополнение существующих моделей на основе анализа логов выполнившихся процессов в этих исследованиях не рассматривается. В то же время фактическое поведение процесса может отличаться от идеального, представленного в модели. Например, в результате применения знаний и опыта исполнителя процесс может быть адаптирован. Поэтому включение в модель гибкого процесса результатов интеллектуального анализа процессов (process mining) позволяет сделать ее более полной.

Общая концепция построения референсной модели гибкого процесса с использованием интеллектуального анализа процессов (process mining) предложена F. Gottschalk [7]. Авторы предложили на основе анализа логов формировать модель, которая охватывает различные варианты поведения процесса. Далее путем конфигурирования из модели исключаются избыточные варианты поведения, нетипичные для данной предметной области. По сравнению с нашим подходом (Compared to our approach, here) в данной работе не рассмотрены вопросы динамической адаптации модели во время выполнения.

Вопросы усовершенствования (Improving) модели процесса представлены M. La Rosa, W.M.P. van der Aalst [8]. Авторы предложили process mining algorithm, который улучшает существующую модель процесса путем дополнения его вариантами выполнения, полученными в результате анализа логов процесса. Важным отличием алгоритма является то, что он использует структурную схожесть (the structural similarity) между существующей моделью и моделью, полученной в результате анализа логов. Подход позволяет построить модель процесса с использованием таких элементов, как parallel execution, choice, and loop execution. Однако для работы данного алгоритма предварительно должна быть задана модель процесса, которая потом усовершенствуется. В отличие от нашего подхода возможность построения модели гибкого процесса полностью на основе анализа логов не рассматривает-

ся. В то же время такая возможность очень важна, поскольку позволяет учесть в модели значительные изменения процесса (например, в результате выполнения реинжиниринга процесса).

B.F. van Dongen и W.M.P. van der Aalst предложен подход, основанный на применении генетического алгоритма при построении дерева процесса (process tree) на основе анализа логов [9]. Согласно данному подходу, модель гибкого процесса может быть представлена в виде дерева, вершины которого отражают связь между действиями процесса (последовательность, параллельное выполнение, выбор, цикл). Однако вопросы построения модели гибкого многовариантного процесса и его конфигурирования на основе адаптации базовых элементов в данной работе не рассматриваются.

M. Rosemann рассмотрел проблему описания конфигурируемой модели гибкого процесса [14]. Рассмотрены коннекторы OR, XOR, AND, а также опциональные связи для конфигурирования модели. Однако, по сравнению с данной работой в описании настраиваемого процесса рассматривается только статическая адаптация. В то же время динамическая адаптация модели при выполнении процесса также является важной.

Обобщая проведенный анализ, следует сказать, что ни один из рассмотренных подходов (In summary, none of the discussed approaches) не позволяет построить модель гибкого процесса на основе анализа логов, отражающих его фактическое поведение, с последующей возможностью как статической, так и динамической адаптации модели.

2. ОБЩИЙ ПОДХОД К ПОСТРОЕНИЮ МОДЕЛИ ГИБКОГО ПРОЦЕССА С ПОМОЩЬЮ ЛОГИЧЕСКИХ СЕТЕЙ

В данной работе предлагается альтернативный традиционным методам process mining [14] подход к получению модели гибкого процесса путем использования методов интеллектуального анализа процессов с применением алгебры конечных предикатов и логических сетей для выделения различных вариантов реализации одного и того же процесса.

В качестве алфавита в алгебре конечных предикатов используются многозначные символы различного назначения: переменных, отношений и кванторов. В рассматриваемом математическом аппарате развиты средства для формульной записи любых конечных отношений, более того, доказано, что данная алгебра является полной и потому позволяет описать произвольные конечные отношения [5].

Решение уравнений АКП осуществляется посредством логической сети, для чего уравнения алгебры преобразуются в систему бинарных предикатов [6]. Каждая логическая сеть создается под заданную модель. При этом алгоритм ее работы остается неизменным, изменяется только состав ее полюсов и ветвей. Модель логической сети задается парой $\langle F(x); G(x, y) \rangle$, состоящей из конечного непустого множества $F(x)$ уравнений алгебры с одной предметной переменной и множества $G(x, y)$ уравнений алгебры с двумя переменными. Уравнения $F(x)$ с одной переменной задают предметные переменные модели, а также области их изменения. Уравнения $G(x, y)$ с двумя переменными задают бинарные отношения между переменными. Первое множество определяет полюса логической сети, а второе – ее ветви.

Ключевое преимущество логических сетей заключается в определении полного набора свойств исследуемого объекта на основе ограниченного входного набора признаков, что обеспечивает возможность эффективного построения модели гибкого процесса на основе анализа логов событий и выделения фрагментов с заданными свойствами.

Фактически в соответствии с предлагаемым подходом при анализе происходит сопоставление логов различных вариантов процессов, выявление отличий с использованием логических сетей, построение объединенного лога без дублирования и противоречий и на этой основе – получение единой модели гибкого процесса традиционными методами анализа процессов.

Предлагаемый подход позволяет объединить базовые способы слияния логов и построения модели процесса:

- объединение моделей различных вариантов реализации процесса, каждая из которых получена традиционными методами process mining [7];
- простое объединение логов для различных вариантов реализации и построение методами process mining единой модели гибкого процесса [7];
- дополнение отличающимися в отдельных реализациях возможностями базовой модели, полученной методами process mining и отражающей типовое, характерное для всех реализаций выполнение процесса [8].

При предлагаемом подходе мы избегаемся от недостатков традиционных методов process mining, сопоставляя и дополняя логи различных вариантов реализации процесса. Выделение отличающихся элементов и проверка непротиворечивости производится с помощью логических сетей.

Отметим, что необходимым условием слияния логов при предлагаемом подходе является соответствие меток (имен) операций в логах, отражающих различные варианты реализации процесса. Однако реализация проверки такого соответствия, а также преобразование имен выходят за рамки данной работы.

Построенная на основе анализа логов модель гибкого процесса не должна содержать тупиков, блокировок и иных дефектных составляющих, что требует выделения базовых элементов модели, а также операторов изменения процесса, обеспечивающих его гибкость.

3. БАЗОВЫЕ ЭЛЕМЕНТЫ МОДЕЛИ ГИБКОГО ПРОЦЕССА

Обобщив предложенную в работах [9, 10] структуризацию, целесообразно выделить следующие основные элементы традиционного, «жесткого» процесса:

- последовательное выполнение действий процесса;
- одновременное (параллельное) выполнение действий;
- циклическое выполнение действий;
- ветвление.

Выбор последующих действий при ветвлении в общем случае определяется состоянием внешней среды процесса.

Отметим, что последний элемент может быть реализован двумя способами – эксклюзивным (XOR) и неэксклюзивным (OR).

Перестройка структуры процесса связана с дополнительными знаниями (например, правилами) о возможности применения тех или иных его опций

(последовательностей действий) в различных условиях. Применение указанных знаний обосновывает выбор между теми или иными ветвями процесса. Остальные ветви не используются (скрываются). Следовательно, в составе гибкого процесса должна быть конструкция, которая позволяет скрыть отдельные ветви (последовательности действий) в модели процесса.

Модель процесса имеет вид системы n бинарных предикатов $M = \{R_i \mid i = \overline{1, n}\}$, задающих правила его поведения. Фактически каждый предикат задает допустимые взаимосвязи между действиями процесса, т. е. состояния являются вершинами, а предикаты – ветвями графа процесса. Если мы ходим скрыть отдельные ветви (предикаты), например R_2 и R_5 , то нужно явно указать, что эти предикаты из модели M удаляются:

$$M = \{R_i \mid i = \overline{1, n}; i \neq 2, 5\}.$$

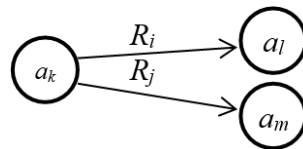
Если необходимо скрыть ветви с заданным общим свойством, то можно задать функцию селекции φ , разделяющую множество предикатов модели M на два непересекающихся подмножества: $R_i \in M'$, если $\varphi(R_i) = R_i$ – предикаты R_i остаются в модели, $i \in \overline{1, n}$; $R_j \in M''$, если $\varphi(R_j) = 0$ – предикаты R_j удаляются из модели, $j \in \overline{1, n}$, $M'' = M \setminus M'$. То есть M' – множество оставшихся предикатов R_i , образующее новую, сокращенную модель, а M'' – множество удаленных предикатов R_j .

При объединении логов нам необходимо, во-первых, распознать ветвление, что означает, что в модели M есть пара предикатов вида

$$M = \begin{cases} \dots \\ R_i(a_k, a_l) \\ \dots \\ R_j(a_k, a_m) \\ \dots \end{cases} \quad (1)$$

Соответствующий фрагмент логической сети представлен на рисунке. Затем, при настройке процесса, ветвление может быть скрыто:

если (1), то или $\varphi(R_i) = R_i$, $\varphi(R_j) = 0$ или $\varphi(R_j) = R_j$, $\varphi(R_i) = 0$.



Ветвление логической сети

Во-вторых, поскольку модель гибкого процесса содержит полный (в общем случае – избыточный) набор допустимых последовательностей операций, то при его настройке может понадобиться пропуск отдельных операций.

Поэтому для гибкого процесса должен быть определен оператор, который обеспечивал бы пропуск отдельных операций или их последовательно-

стей. Тогда формально систему предикатов P , реализующую последовательность операций,

$$P = \begin{cases} R_1(a_1, a_2), \\ R_2(a_2, a_3), \\ \dots \\ R_n(a_n, a_{n+1}), \end{cases} \quad (2)$$

нужно заменить одним предикатом $P'(a_1, a_{n+1})$.

При объединении логов необходимо распознать подобные последовательности, содержащие полные и сокращенные наборы действий. Критерием подобия пары последовательностей операций является выполнение двух условий:

1) в первых предикатах первой и второй последовательностей первым аргументом является одна и та же переменная;

2) в последних предикатах первой и второй последовательностей вторым аргументом является одна и та же переменная.

Последовательностью может быть и единственный предикат. Схематически пример пары подобных последовательностей операций P_1 и P_2 можно представить так:

$$P_1 : \quad (a_1) \xrightarrow{R_1} (a_2) \xrightarrow{R_2} (a_3)$$

$$P_2 : \quad (a_1) \xrightarrow{R_3} (a_3).$$

Оператор пропуска (2) позволяет переходить от полного к сокращенному набору.

В-третьих, сокрытие отдельных ветвей процесса и пропуск операций приводят к изменению последовательности действий процесса. Отсюда возникает задача изменения последовательности действий, что может быть решено с помощью соответствующего оператора. Однако произвольная перестановка операций изменяет модель процесса. Поэтому мы ограничимся переключением между ограниченным набором возможных последовательностей операций, удовлетворяющих признаку подобия, описанному выше.

При объединении логов первоначально выполняется формирование конечных наборов возможных последовательностей для каждой пары действий, имеющей более одного варианта последовательностей операций от первого действия к последнему. Такие наборы формируются попарным сравнением по признаку подобия.

Затем оператор изменения последовательности действий позволяет переходить от одного набора к другому из множества допустимых:

$$P^* \in \{P_1, P_2, \dots, P_n\}. \quad (3)$$

В-четвертых, избыточность полной модели процесса можно уменьшить путем упрощения базовых действий. Рассмотрим такое упрощение по отдельным элементам процесса.

Последовательное выполнение действий процесса может быть строгим и нестрогим. При нестрогой последовательности второе действие может начинаться после начала первого, т. е. допустимы варианты P_1 и P_2 фрагментов модели:

$$P_1: (a_1) \xrightarrow{R_1} (a_2) \xrightarrow{R_2} (a_3)$$

$$P_2: (a_1) \xrightarrow{R_3} (a_2) \xrightarrow{R_2} (a_3).$$

Однако первое не может быть начато позже начала второго. Это условие можно учесть, введя в предикатную модель (или логическую сеть) дополнительный предикат второго порядка T , задающего допустимую последовательность выполнения каждого действия (предиката модели):

$$T(P_1, P_2, \dots, P_n). \quad (4)$$

При строгой последовательности накладывается дополнительное ограничение: выполнение второго действия допускается только после полного завершения первого. Упрощение данного элемента сводит нестрогую последовательность к строгой. Формально это задается таблицей истинности предиката T .

Параллельное выполнение действий процесса фактически расширяет нестрогую последовательность, поскольку для пары параллельных действий второе действие может начаться после начала первого, так же и первое может начаться после начала второго. Очевидно, что параллельное выполнение упрощается до нестрогого последовательного и затем до строгой последовательности.

Циклическое выполнение действий предусматривает многократный повтор одной и той же последовательности. При этом задается либо количество повторов, либо условие останова. Поэтому упрощение цикла ограничивает количество повторений либо сводит цикл к одному повторению.

Контроль циклического выполнения действий можно реализовать введением в предикатную модель (логическую сеть) еще одного вида предиката второго порядка S_i для каждого i -го цикла. Аргументами предикатов второго порядка S_i являются, кроме предикатов i -го цикла, еще две переменные: n – количество повторов, z – условие останова цикла.

Нестрогая последовательность внутри цикла также сводится к строгой, как описано выше.

Последний базовый элемент процесса – выбор, исключаящий (XOR) или неисключающий (OR). Неисключающий выбор упрощается сведением его от оператора «ИЛИ» к оператору «И».

Дальнейшее возможное упрощение – сведение к последовательности операций.

Исключающий выбор сводится непосредственно к последовательности операций выбором одной из возможных ветвей процесса. Приведенные операторы обеспечивают статическую и динамическую настройку модели. Взаимодействие элементов процесса и операторов, обеспечивающих его гибкость, представлено в табл. 1 и 2.

Таблица 1

Динамическая настройка: результаты применения операторов адаптации к базовым элементам процесса

Операторы	Элементы процесса			
	Последовательное выполнение	Параллельное выполнение	Циклическое выполнение	Выбор действий
Динамическая настройка				
Скрытие выбора	–	–	–	Ограниченный выбор
Пропуск операций	+	+	+	Вместо данного оператора используется ограничение выбора
Изменение последовательности	Смена порядка выполнения операций	–	–	–

Таблица 2

Статическая настройка: результаты применения операторов адаптации к базовым элементам процесса

Операторы	Элементы процесса			
	Последовательное выполнение	Параллельное выполнение	Циклическое выполнение	Выбор действий
Адаптация модели к предметной области	Нестрогая последовательность преобразуется в строгую	Преобразуется в нестрогую, а затем в строгую последовательность	Задается ограниченное количество повторений и преобразуется в нестрогую, а затем в строгую последовательность	<ol style="list-style-type: none"> 1. Выбор по XOR преобразуется в последовательность операций выбором одной из ветвей 2. Выбор по OR преобразуется к совместному выбору всех ветвей и затем к их последовательному выполнению

ЗАКЛЮЧЕНИЕ

В работе предложен подход к построению модели гибкого процесса методами интеллектуального анализа процессов с применением логических сетей. Для реализации данного подхода рассмотрен список типовых элементов и предложены операторы настройки элементов гибкого процесса. Главное отличие предложенного подхода состоит в том, что применение операторов настройки позволяет адаптировать модель гибкого процесса как статически, так и динамически. В первом случае адаптация выполняется с учетом известных статических характеристик предметной области до начала использования процесса. Во втором случае изменение процесса осуществляется в зависимости от знаний о его окружении во время его использования. Также при динамической адаптации модели для усовершенствования процесса могут учитываться знания и опыт персонала, выполняющего процесс. Автор признателен д-ру техн. наук А.В. Чехонадских за полезные замечания.

СПИСОК ЛИТЕРАТУРЫ

1. *Aalst W.M.P. van der*. Process mining: discovery, conformance and enhancement of business processes. – Berlin Heidelberg: Springer-Verlag, 2011. – 352 p. – doi: 10.1007/978-3-642-19345-3.
2. *Koryukin A.N., Chekhonadskikh A.V.* Extreme root location of real polynomials and stabilization of 3-mass control system // Algebra and Model Theory 8: Collection of papers / Novosibirsk State Technical University. – Novosibirsk: NSTU Publ., 2011. – P. 19–39.
3. *Воевода А.А., Корюкин А.Н., Чехонадских А.В.* О понижении порядка стабилизирующего управления на примере двойного перевернутого маятника // Автометрия. – 2012. – Т. 48, № 6. – С. 69–83.
4. *Чехонадских А.В.* Экстремальные расположения полюсов систем автоматического управления с регулятором пониженного порядка // Автоматика и телемеханика. – 2014. – № 10. – С. 6–24.
5. *Бондаренко М.Ф., Шабанов-Кушнаренко Ю.П.* Теория интеллекта: учебник. – Харьков: СМИТ, 2007. – 576 с.
6. *Бондаренко М.Ф., Шабанов-Кушнаренко Ю.П.* Уравнения теории интеллекта // Бионика интеллекта. – 2011. – № 3 (77). – С. 30–45.
7. *Gottschalk F., Aalst W.M.P. van der, Jansen-Vullers M.H.* Mining reference process models and their configurations // On the Move to Meaningful Internet Systems: OTM 2008 Workshops (Monterrey, Mexico, November 9–14, 2008) / Ed. R. Meersman, Z. Tari, P. Herrero. – Berlin; Heidelberg: Springer-Verlag, 2008. – P. 263–272. – (Lecture Notes in Computer Science; vol. 5333). – doi: 10.1007/978-3-540-88875-8_47.
8. Improving business process models using observed behavior / J.C.A.M. Buijs, M. La Rosa, H.A. Reijers, B.F. Dongen, W.M.P. van der Aalst // Proceedings of the Second International Symposium on Data-Driven Process Discovery and Analysis: Second IFIP WG 2.6, 2.12 International Symposium, SIMPDA 2012, Campione d'Italia, Italy, June 18–20, 2012, Revised Selected Papers. – Berlin; Heidelberg: Springer-Verlag, 2013. – P. 44–59. – (Lecture Notes in Business Information Processing; vol. 162). – doi: 10.1007/978-3-642-40919-6_3.
9. *Buijs J.C.A.M., Dongen B.F. van, Aalst W.M.P. van der.* On the role of fitness, precision, generalization and simplicity in process discovery // On the Move to Meaningful Internet Systems: OTM 2012: Confederated International Conferences: CoopIS, DOA-SVI, and ODBASE 2012, Rome, Italy, September 10–14, 2012. Proceedings. – Berlin; Heidelberg: Springer-Verlag, 2012. – Pt. 1. – P. 305–322. – (Lecture Notes in Computer Science; vol. 7565). – doi: 10.1007/978-3-642-33606-5_19.
10. *Rosemann M., Aalst W.M.P. van der.* A configurable reference modeling language // Information Systems. – 2007. – Vol. 32, iss. 1. – P. 1–23. – doi: 10.1016/j.is.2005.05.003.

11. *Ретин В.В., Елиферов В.Г.* Процессный подход к управлению. Моделирование бизнес-процессов. – М.: Стандарты и качество, 2008. – 404 с. – (Практический менеджмент).
12. *Gottschalk F., Aalst W.M.P. van der, Jansen-Vullers M.H.* Merging event-driven process chains // On the Move to Meaningful Internet Systems: OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008, Monterrey, Mexico, November 9–14, 2008. Proceedings. – Berlin; Heidelberg: Springer-Verlag, 2008. – Pt. 1. – P. 418–426. – (Lecture Notes in Computer Science; vol. 5331). – doi: 10.1007/978-3-540-88871-0_28.
13. *Dongen B.F. van, Aalst W.M.P. van der.* Multi-phase mining: aggregating instances graphs into EPCs and Petri nets // Proceedings of the Second International Workshop on Applications of Petri Nets to Coordination, Workflow and Business Process Management, Florida International University. – Miami, Florida, USA, 2005. – P. 35–58.
14. *Uchitel S., Chechik M.* Merging partial behavioural models // ACM SIGSOFT Software Engineering Notes. – 2004. – Vol. 29, iss. 6. – P. 43–52. – doi: 10.1145/1041685.1029904.
15. Business process model merging: an approach to business process consolidation / M. La Rosa, M. Dumas, R. Uba, R. Dijkman // ACM Transactions on Software Engineering and Methodology. – 2012. – Vol. 22, iss. 2. – Art. 11. – doi: 10.1145/2430545.2430547.
16. Similarity of business process models: metrics and evaluation / R. Dijkman, M. Dumas, B. van Dongen, R. Uba, J. Mendling // Information Systems. – 2011. – Vol. 36, iss. 2. – P. 498–516. – doi: 10.1016/j.is.2010.09.006.
17. Merging business process models / M. La Rosa, M. Dumas, R. Uba, R. Dijkman // On the Move to Meaningful Internet Systems: Confederated International Conferences: CoopIS, IS, DOA and ODBASE, Hersonissos, Crete, Greece, October 25–29, 2010, Proceedings. – Berlin; Heidelberg: Springer, 2010. – Pt. 1. – P. 96–113. – (Lecture Notes in Computer Science; vol. 6426). – doi: 10.1007/978-3-642-16934-2_10.
18. *Reijers H.A., Mans R.S., Toorn R.A. van der.* Improved model management with aggregated business process models // Data and Knowledge Engineering. – 2009. – Vol. 68, iss. 2. – P. 221–243. – doi: 10.1016/j.datak.2008.09.004.
19. *Rosemann M., Aalst W.M.P. van der.* A configurable reference modelling language // Information Systems. – 2007. – Vol. 32, iss. 1. – P. 1–23. – doi: 10.1016/j.is.2005.05.003.
20. *Dongen B. van, Dijkman R., Mendling J.* Measuring similarity between business process models // Advanced Information Systems Engineering: 20th International Conference, CAiSE 2008, Montpellier, France, June 16–20, 2008: Proceedings. – Berlin; Heidelberg: Springer-Verlag, 2008. – P. 450–464. – (Lecture Notes in Computer Science; vol. 5074). – doi: 10.1007/978-3-540-69534-9_34.
21. *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. – Vol. 19, iss. 3–4. – P. 159–203. – doi: 10.1142/S0218843010002139.
22. *Li C., Reichert M., Wombacher A.* Mining business process variants: challenges, scenarios, algorithms // Data & Knowledge Engineering. – 2011. – Vol. 70, iss. 5. – P. 409–434. – doi: 10.1016/j.datak.2011.01.005.
23. *Li C., Reichert M.U., Wombacher A.* A heuristic approach for discovering reference models by mining process model variants: technical report TR-CTIT-09-08 / University of Twente, Centre for Telematics and Information Technology. – Enschede, Netherlands, 2009. – 51 p. – ISSN 1381-3625.
24. *Wombacher A., Li C.* Alternative approaches for workflow similarity // IEEE International Conference on Service Computing (SCC 2010), 5–10 July 2010. – Miami, Florida, USA, 2010. – P. 337–345. – doi: 10.1109/SCC.2010.95.

Рудометкина Моника Николаевна, аспирант Томского политехнического университета. Основное направление научных исследований – теория искусственного интеллекта, логические сети. Имеет более 20 публикаций. E-mail: mn.rud@inbox.ru

*Adaptation of basic elements in designing a flexible process model**

M.N. RUDOMETKINA

National Research Tomsk Polytechnic University, 30 Lenin Prospekt, Tomsk, 634050, Russian Federation, postgraduate. E-mail: mn.rud@inbox.ru

The work is based on the idea of constructing a model of a flexible process through the use of process mining techniques with the use of algebra of finite predicates (AFP) and logical networks. Algebra of finite predicate is intended to formalize logical nature processes and is discrete formalism for describing arbitrary finite relations. Solving equations of AFP is implemented through a logical network and to do this algebraic equations are converted into a binary system.

The proposed approach consists in successive complementing of the model at the stage of the log analysis for various versions of the process. Identification of different elements is performed using logical nets.

The flexibility of the process model is achieved by applying operators to adapt the basic elements of the model. As the model of a flexible process contains all possible options for its implementation, adaptation is usually used to eliminate excess capacity. Serial, parallel or looping execution and a follow-up selection process are used as basic elements of the model.

Static adaptation is performed at the model configuration stage in the information system and takes into account features of the subject area. Dynamic adaptation is carried out in implementing the process model in case external environment conditions of the process change.

Keywords: intelligent analysis, logical networks, flexible process, logs events, algorithms, models, relations, operator adaptation process, static adaptation, dynamic adaptation

DOI: 10.17212/1814-1196-2015-1-107-120

REFERENCES

1. Aalst W.M.P. van der. *Process mining: discovery, conformance and enhancement of business processes*. Berlin Heidelberg: Springer-Verlag, 2011. 352 p. doi: 10.1007/978-3-642-19345-3
2. Koryukin A.N., Chekhonadskikh A.V. Extreme root location of real polynomials and stabilization of 3-mass control system. *Algebra and Model Theory* 8. Novosibirsk, NSTU Publ., 2011, pp. 19–39.
3. Voevoda A.A., Koryukin A.N., Chekhonadskikh A.V. O ponizhenii poryadka stabiliziruyushchego upravleniya na primere dvoynogo perevernutogo mayatnika [Reducing the stabilizing control order for a double inverted pendulum]. *Avtometriya – Optoelectronics, Instrumentation and Data Processing*, 2012, vol. 48, iss. 6, pp. 593–604. doi: 10.3103/S8756699012060076. Translated from *Avtometriya*, 2012, vol. 48, no. 6, pp. 69–83.
4. Chekhonadskikh A.V. Ekstremal'nye raspolozheniya polyusov sistem avtomaticheskogo upravleniya s regulyatorom ponizhennogo poryadka [Extremal pole placement in control systems with a low order controller]. *Avtomatika i telemekhanika – Automation and Remote Control*, 2014, vol. 75, iss. 10, pp. 1717–1731. Translated from *Avtomatika i telemekhanika*, 2014, no. 10, pp. 6–24.
5. Bondarenko M.F., Shabanov-Kushnarenko Yu.P. *Teoriya intellekta* [Theory of intelligence]. Khar'kov, SMIT Publ., 2007. 576 p.
6. Bondapenko M.F., Shabanov-Kushnarenko Yu.P. *Uraveniya teorii intellekta* [The intellect theory equations]. *Bionika intellekta – Bionics of Intelligence*, 2011, no. 3 (77), pp. 30–45.
7. Gottschalk F., Aalst W.M.P. van der, Jansen-Vullers M.H. Mining reference process models and their configurations. *Lecture Notes in Computer Science: On the Move to Meaningful Internet Systems: OTM 2008 Workshops* (Monterrey, Mexico, November 9–14, 2008). Berlin, Heidelberg, Springer-Verlag, 2008, vol. 5333, pp. 263–272. doi: 10.1007/978-3-540-88875-8_47
8. Buijs J.C.A.M., La Rosa M., Reijers H.A., Dongen B.F., Aalst W.M.P. van der. Improving business process models using observed behavior. *Lecture Notes in Business Information Processing: Data-Driven Process Discovery and Analysis: proceedings of the Second IFIP WG 2.6, 2.12 Interna-*

* Received 28 August 2014.

This work is supported by the Grant of Russian Foundation for Basic Research № 12-08-00296.

tional Symposium, SIMPDA 2012, Campione d'Italia, Italy, June 18–20, 2012, Revised Selected Papers. Berlin, Heidelberg, Springer-Verlag, 2013, vol. 162, pp. 44–59. doi: 10.1007/978-3-642-40919-6_3

9. Buijs J.C.A.M., Dongen B.F. van, Aalst W.M.P. van der. On the role of fitness, precision, generalization and simplicity in process discovery. *Lecture Notes in Computer Science: On the Move to Meaningful Internet Systems: OTM 2012: Confederated International Conferences: CoopIS, DOA-SVI, and ODBASE 2012*, Rome, Italy, September 10–14, 2012. Proceedings. Berlin, Heidelberg, Springer-Verlag, 2012, vol. 7565, pt. 1, pp. 305–322. doi: 10.1007/978-3-642-33606-5_19

10. Rosemann M., Aalst W.M.P. van der. A configurable reference modeling language. *Information Systems*, 2007, vol. 32, iss. 1, pp. 1–23. doi: 10.1016/j.is.2005.05.003

11. Repin V.V., Eliferov V.G. *Protsessnyi podkhod k upravleniyu. Modelirovanie biznes-protsessov* [Process approach to management. Business processes simulation]. Moscow, Standarty i kachestvo Publ., 2008. 404 p.

12. Gottschalk F., Aalst W.M.P. van der, Jansen-Vullers M.H. Merging event-driven process chains. *Lecture Notes in Computer Science: On the Move to Meaningful Internet Systems: OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008*, Monterrey, Mexico, November 9–14, 2008. Proceedings. Berlin, Heidelberg, Springer-Verlag, 2008, vol. 5331, pt. 1, pp. 418–426. doi: 10.1007/978-3-540-88871-0_28

13. Dongen B.F. van, Aalst W.M.P. van der. Multi-phase mining: aggregating instances graphs into EPCs and Petri nets. *Proceedings of the Second International Workshop on Applications of Petri Nets to Coordination, Workflow and Business Process Management*. Miami, Florida, USA, Florida International University, Miami, Florida, USA, 2005, pp. 35–58.

14. Uchitel S., Chechik M. Merging partial behavioural models. *ACM SIGSOFT Software Engineering Notes*, 2004, vol. 29, iss. 6, pp. 43–52. doi: 10.1145/1041685.1029904

15. 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, vol. 22, iss. 2, art. 11. doi: 10.1145/2430545.2430547

16. Dijkman R., Dumas M., Dongen B. van, Uba R., Mendling J. Similarity of business process models: metrics and evaluation. *Information Systems*, 2011, vol. 36, iss. 2, pp. 498–516. doi: 10.1016/j.is.2010.09.006

17. La Rosa M., Dumas M., Uba R., Dijkman R. Merging business process models. Merging business process models. *Lecture Notes in Computer Science: On the Move to Meaningful Internet Systems: Confederated International Conferences: CoopIS, IS, DOA and ODBASE, Hersonissos, Crete, Greece, October 25–29, 2010*. Proceedings. Berlin, Heidelberg, Springer, 2010, vol. 6426, pt. 1, pp. 96–113. doi: 10.1007/978-3-642-16934-2_10

18. Reijers H.A., Mans R.S., Toorn R.A. van der. Improved model management with aggregated business process models. *Data and Knowledge Engineering*, 2009, vol. 68, iss. 2, pp. 221–243. doi: 10.1016/j.datak.2008.09.004

19. Rosemann M., Aalst W.M.P. van der. A configurable reference modelling language. *Information Systems*, 2007, vol. 32, iss. 1, pp. 1–23. doi: 10.1016/j.is.2005.05.003

20. Dongen B. van, Dijkman R., Mendling J. Measuring similarity between business process models. *Lecture Notes in Computer Science: Advanced Information Systems Engineering: 20th International Conference, CAiSE 2008, Montpellier, France, June 16–20, 2008: Proceedings*. Berlin, Heidelberg: Springer-Verlag, 2008, vol. 5074, pp. 450–464. doi: 10.1007/978-3-540-69534-9_34

21. 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, vol. 19, iss. 3–4, pp. 159–203. doi: 10.1142/S0218843010002139

22. Li C., Reichert M., Wombacher A. Mining business process variants: challenges, scenarios, algorithms. *Data & Knowledge Engineering*, 2011, vol. 70, iss. 5, pp. 409–434. doi: 10.1016/j.datak.2011.01.005

23. Li C., Reichert M.U., Wombacher A. *A heuristic approach for discovering reference models by mining process model variants*: technical report TR-CTIT-09-08. Enschede, Netherlands, University of Twente, Centre for Telematics and Information Technology, 2009. 51 p. ISSN 1381-3625

24. Wombacher A., Li C. Alternative approaches for workflow similarity. *IEEE International Conference on Service Computing (SCC 2010)*, Miami, Florida, USA, 5–10 July 2010, pp. 337–345. doi: 10.1109/SCC.2010.95