

8. Константинов В.Н., Абдрахманов Р.С. Оценка производительности ветроэнергетической установки с помощью функции распределения Вейбулла // Известия вузов: Проблемы энергетики. – 2006 – №11-12. – С. 76-79.

9. Голдаев С.В., Радюк К.Н. Проверка с помощью критерия Пирсона статистической гипотезы о распределении генеральной совокупности по закону Вейбулла-Гнеденко // Материалы двадцатой Всероссийской научно-технической конференции «Энергетика: экология, надежность, безопасность». – Томск: Изд. ТПУ, 2014. – С. 127-130.

10. Кобзарь А. И. Прикладная математическая статистика. Для инженеров и научных работников. – М.: Физматлит, 2006. – 816 с.

11. Обухов С.Г., Сурков М.А., Хошнау З.П. Методика выбора ветроэнергетических установок малой мощности // Электро. – 2011. – №2. – С. 25-30.

БАЗОВЫЕ ЭЛЕМЕНТЫ ПРИ ПОСТРОЕНИИ МОДЕЛИ ГИБКОГО ПРОЦЕССА

М.Н. Рудометкина, аспирант,

П.А. Каковкин, магистрант гр. 8ВМ41

Томский политехнический университет, 634050, г.Томск, пр.Ленина,30,

E-mail: mn.rud@inbox.ru

Интеллектуальный анализ процессов (process mining) предназначен для построения моделей информационных процессов преобразования ресурсов [1; 5-7; 1-14]. 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 графо-вые 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:

- скрытые неформализованные знания о выполнении процесса, влияющие на последовательность его активностей;
- опыт исполнителей процесса, приводящий к изменению внутренней структуры активностей и, следовательно, изменяющий их результат;
- порядок взаимодействия (часто неформальный) исполнителей – людей либо организаций при выполнении процесса;
- территориальная распределенность процесса, влияющая 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 жёсткого упорядочения. Её 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.

Базовые элементы модели гибкого процесса. Обобщив предложенную в работах [7, 8] 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 – эксклюзивный (XOR) и не эксклюзивный (OR) выбор.

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, $i \in \overline{1, n}$ и $R_j \in \bar{M}$, если $\varphi(R_j) = 0$ - предикаты R_j удаляются из модели, $j \in \overline{1, n}$, $M = M \setminus \bar{M}'$. Т.е. M' – множество оставшихся предикатов R_i , образующее новую, сокращенную модель, а \bar{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 рис. 1.

Затем, при настройке процесса, ветвление может быть скрыто: если (1), то или $\varphi(R_i) = R_i$, $\varphi(R_j) = 0$ или $\varphi(R_j) = R_j$, $\varphi(R_i) = 0$.

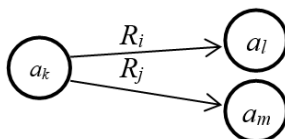


Рис. 1. Ветвление логической сети.

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 последовательностей. 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:

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

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

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

В-третьих, сокрытие отдельных ветвей процесса и пропуск операций приводит 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, также 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 C_i для каждого i -го цикла. Аргументами предикатов второго порядка C_i являются, кроме предикатов i -го цикла, еще две переменных: n – количество повторов, z – условие останова цикла. Нестрогая последовательность внутри цикла также сводится к строгой, как описано выше.

Последний базовый элемент процесса – выбор, исключающий (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.

Выводы. 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. Также при динамической адаптации модели для усовершенствования процесса могут учитываться знания и опыт персонала, выполняющего процесс. Автор признателен д.т.н. А.В. Чехонадских за полезные замечания.

Список литературы:

1. W.M.P. van der Aalst. Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer. – 2011. – 370 p.
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/ Novos. State Techn. Univ.– Novosibirsk. – 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. F. Gottschalk, W.M.P. van der Aalst, and M.H. Jansen-Vullers. Mining Reference Process Models and their Configurations. OTM 2008 Workshops.– volume 5333 of Lecture Notes in Computer Science/Berlin Heidelberg. –Springer Verlag. – 2008. – P. 263–272.
6. J.C.A.M. Buijs, M. La Rosa, H.A. Reijers, B.F. Dongen, and W.M.P. van der Aalst. Improving Business Process Models using Observed Behavior// In Proceedings of the Second Intern. Symposium on Data-Driven Process Discovery and Analysis.– LNBP. Springer. – 2013.
7. 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 Proceedings of Coop IS. LNCS. Springer. – 2012.

8. M. Rosemann, W.M.P. van der Aalst. A Configurable Reference Modeling Language. Information Systems, Т.1. –№32. Репин В., Елифиров В. Процессный подход к управлению. Моделирование бизнес-процессов. М.: Стандарты и качество. – 2008. – 196 с.
9. F. Gottschalk, W.M.P. van der Aalst, M.H. Jansen-Vullers. Merging Event-driven Process Chains. OTM 2008, Part I, CoopIS, volume 5331 of Lecture Notes in Computer Science. Berlin Heidelberg. Springer Verlag. – 2008.– P. 418–426.
10. B. van Dongen, W.M.P. van der Aalst. Multi-Phase Mining: Aggregating Instances Graphs into EPCs and Petri Nets. In Proceedings of the Second International Workshop on Applications of Petri Nets to Coordination, Workflow and Business Process Management. Florida International University.– Miami. FL. USA. – 2005. – P. 35–58.
11. S. Uchitel, M. Chechik. Merging Partial Behavioural Models. SIGSOFT Software Engineering Notes. – 29 (6). – 2004. – P. 43–52.
12. La Rosa, M. Dumas, R. Uba, R. Dijkman. Business Process Model Merging: An Approach to Business Process Consolidation. ACM Transactions on Software Engineering and Methodology. – 2012. – № 22 (2).
13. Dijkman, R., Dumas, M., van Dongen, B., Uba, R., and Mendling, J. Similarity of business process models: Metrics and evaluation. Information Systems. – 2011. V. 36 (2). – P. 498–516.
14. La Rosa M., Dumas M., Uba R., Dijkman. Merging business process models. In Proceedings of CoopIS. LNCS. – Springer Reijers – 2010.

МОДЕЛИРОВАНИЕ УЛИЧНО-ДОРОЖНОЙ СЕТИ КИРОВСКОГО РАЙОНА ГОРОДА ТОМСКА СРЕДСТВАМИ ППП «ANYLOGIC 7»

В.Г. Ротарь, к.т.н., доц.,

Н.А. Стекленин, магистрант гр. 8КМ31

Томский политехнический университет, 634050, г.Томск, пр.Ленина,30,

тел. (923)-612-34-94

E-mail: nas.yrg@gmail.com

Введение. В последние десятилетия во многих крупных городах исчерпаны или близки к исчерпанию возможности экстенсивного развития транспортных сетей. Конечно один из вариантов решения проблемы – это полная модернизация и реконструкция улично – дорожной сети города, однако стоимость данного проекта не по карману городскому бюджету. Поэтому особую важность приобретает оптимальное планирование сетей, улучшение организации движения, оптимизация системы маршрутов общественного транспорта [1]. Решение таких задач невозможно без математического моделирования транспортных сетей. Главная задача математических моделей – определение и прогноз всех параметров функционирования транспортной сети, таких как интенсивность движения на всех элементах сети, средние скорости движения, задержки и потери времени и т.д.

Выбор среды моделирования. Транспортные модели делятся на математические и имитационные. Первые оперируют известными законами движения транспорта, представленными в виде формул, систем уравнений и т.п.