

УДК 004.067

С. А. Аббас

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Модели бизнес-процессов для систем мониторинга

Современный этап развития техники и технологии приводит к появлению систем принципиально нового уровня сложности, которая выражается в увеличении числа элементов и в усложнении структур реализуемых бизнес-процессов. Для определения состояния сложных технических систем используются разнообразные системы мониторинга, в частности использующие алгоритмы Process Mining. Однако современные системы Process Mining в значительной степени ориентированы на простые модели бизнес-процессов, использующие только потоки управления. Для решения реальных задач требуется кроме потоков управления строить модели, учитывающие потоки данных и ресурсы. В статье предлагается новая модель бизнес-процессов, основанная на стратегиях вычислений, и рассматриваются 16 различных стратегий управления. Использование предлагаемого подхода позволяет решить ряд важных задач, связанных с управлением бизнес-процессами, которые не могут быть решены с помощью традиционных моделей, основанных на использовании моделей потоков управления.

Бизнес-процессы, система мониторинга, интернет вещей

Задача мониторинга бизнес-процессов является одной из подзадач, решаемой в рамках систем управления бизнес-процессами (БП) [1]. Основное назначение систем мониторинга (СМ) состоит в получении оперативной и адекватной информации о состоянии наблюдаемой системы (НС), которая в общем случае является распределенной. Классические подходы к построению СМ в значительной степени исходят из того, что структура НС статична, т. е. изменяется достаточно редко и при этом имеется достаточно полная информация обо всех изменениях в ее структуре. Для управления подобными сетевыми структурами достаточно успешно можно применять системы управления, ориентированные на стандарт ITSM (Information Technology Service Management) [2]. Однако современные системы, в частности системы, построенные по принципу интернета вещей, в основе своей – это системы с постоянно изменяющейся (динамической) структурой и сложным поведением, и применение к ним классических подходов не приносит желаемых результатов.

Модельный подход к решению задачи мониторинга. К возможным подходам для решения задачи мониторинга относится модельный, предполагающий активное использование моделей процессов и структур НС [3]. Наиболее широко

известен подход, известный как Process Mining [4], в рамках которого наряду с построением моделей бизнес-процессов в форме графа потока работ определяется соответствие реального бизнес-процесса и его модели, представленной в подобной форме. В этом направлении достигнуты впечатляющие результаты, однако имеются и проблемы. Основные проблемы связаны с двумя моментами: используемые модели потока работ ориентированы на описание относительно простых структур БП или на статическую структуру БП. В этом плане представляется целесообразным рассмотреть возможность использования более сложных моделей для описания БП.

Анализ требований, предъявляемых к моделям БП систем мониторинга. Речь идет только о машинно-исполняемых БП, прочие выходят за рамки данной статьи.

Основные требования, предъявляемые к моделям:

1. Возможность описания многоуровневых параллельных исполняемых БП в терминах потока управления, потока данных, потока запросов и ресурсов.

2. Возможность достаточно простого перехода от моделей к исполняемому коду для режимов как трансляции, так и интерпретации.

3. Возможность автоматического построения многоуровневой модели (синтеза) по логам.

4. Возможность моделировать различные механизмы управления БП.

Похоже, что для разных задач нужны разные модели, а значит для того чтобы покрыть все задачи, нужно семейство моделей, т. е. требуется разработать метамодель.

Автору не удалось найти ни одной модели, удовлетворяющей всем перечисленным требованиям. Наиболее близкими можно считать теорию акторов, разработанную в МТИ [5], и модель, основанную на стратегиях управления, разработанную в ЛЭТИ [6]. Обе модели были разработаны независимо друг от друга еще в середине 1980-х гг., но до сих пор не потеряли актуальности.

Акторная модель изначально предназначалась не для использования в качестве метамодели, а для построения на ее основе языка, способного описать распределенные вычисления [5], хотя следует отметить, что модель акторов может использоваться в качестве метамодели. Частные модели могут быть получены посредством формирования профилей, однако применительно к построению систем мониторинга этот подход представляется излишне сложным.

Подход, основанный на стратегиях управления БП, подходит больше, поскольку изначально он предлагался в качестве метамодели вычислений и концентрируется на механизмах управления БП. Этот подход в значительной степени также был ориентирован на построение исполняемых БП и доменно-ориентированных языков, однако после соответствующей доработки он может быть взят за основу. Поскольку напрямую данный подход не ориентирован на описание БП с целью их мониторинга, в модифицированный подход должны быть включены элементы, которые представляются важными применительно к задаче построения систем мониторинга. В первую очередь речь идет об описании многоуровневых БП и наделении акторов способностью порождать другие акторы.

Таким образом, для описания БП предлагается использовать модифицированную модель стратегий управления.

Метамодель БП. Метамодель предназначена прежде всего для описания машинно-исполняемых БП с точки зрения системы мониторинга.

Метамодель БП можно определить следующим образом: $MM = \{O, R, STR\}$, где O – операторы; R – ресурсы; STR – стратегии управления БП. В свою очередь, операторы можно определить как $O = \{O_t, O_m, O_p, O_c, O_{mod}\}$, где O_t – операторы трансформации; O_m – операторы работы с ДИЗ (хранение, поиск, извлечение); O_c – операторы управления; O_{mod} – операторы-модели. Ресурсные акторы можно определить следующим образом: $R = \{R_p, R_{tn}, R_e, R_g, \dots\}$, где R_p – актор выделения ресурса (pick up); R_{tn} – актор настройки ресурса (tuning); R_e – актор занятия ресурса (engage); R_g – актор предоставления ресурса. Кроме того, могут использоваться и другие типы ресурсных акторов, например $R_{tn e}$ – актор настройки и занятия ресурса.

Стратегию можно определить как высокоуровневое описание способа управления БП в терминах сторожевых условий запуска операторов $STR = \{GR_i\}$, где GR_i – i -й способ (guard condition, спусковая функция, триггер) определения условий запуска операторов. В самом общем виде условия запуска некоторого оператора в рамках какой-либо стратегии можно определить как $GR = GR_{ctr} \wedge GR_{dat} \wedge GR_{res} \wedge GR_{req}$, где GR_{ctr} – сигнал, предписывающий начало выполнения оператора; GR_{da} – готовность данных; GR_{res} – наличие ресурсов, GR_{req} – наличие запроса на выполнение оператора.

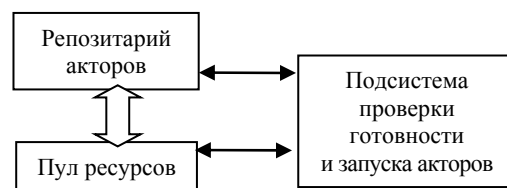
Можно определить вектор $V_{GR} = \{GF_{ctr}, GF_{dat}, GF_{res}, GF_{req}\}$, где GF – использование отдельных механизмов определения готовности акторов, который может принимать значение 0 (условие не проверяется) или 1 (условие проверяется). Очевидно, что все условия должны выполняться тем или иным способом. Это может происходить по-разному: в статике (на этапе компиляции) в динамике, централизованно, распределенно. В данном случае под термином *проверка условия* понимается реализация проверки соответствующего условия в динамике (run time), т. е. условия готовности проверяются на этапе выполнения. Таким образом, можно определить 16 различных стратегий управления БП, характеристики которых приведены в таблице.

№	V_{GR}	Название	Условное обозначение	Проверяемые условия
1	0000	Вырожденная	nil	$GR = 1$
2	0001	Запросная	Z	$GR = GF_{req}$
3	0010	Ресурсная	R	$GR = GF_{res}$
4	0011	Запросно-ресурсная	ZR	$GR = GF_{res} \wedge GF_{req}$
5	0100	Потоковая	D	$GR = GF_{dat}$
6	0101	Потоково-запросная	DZ	$GR = GF_{dat} \wedge GF_{req}$
7	0110	Потоково-ресурсная	DR	$GR = GF_{dat} \wedge GF_{res}$
8	0111	Потоково-запросно-ресурсная	DZR	$GR = GF_{dat} \wedge GF_{req} \wedge GF_{res}$
9	1000	Директивная	C	$GR = GR_{ctr}$
10	1001	Директивно-запросная	CZ	$GR = GR_{ctr} \wedge GF_{req}$
11	1010	Директивно-ресурсная	CR	$GR = GR_{ctr} \wedge GF_{res}$
12	1011	Директивно-запросно-ресурсная	CZR	$GR = GR_{ctr} \wedge GF_{req} \wedge GF_{res}$
13	1100	Директивно-потоковая	CD	$GR = GR_{ctr} \wedge GF_{dat}$
14	1101	Директивно-потоково-запросная	CDZ	$GR = GR_{ctr} \wedge GF_{req}$
15	1110	Директивно-потоково-ресурсная	CDR	$GR = GR_{ctr} \wedge GF_{dat} \wedge GF_{res}$
16	1111	Избыточная	$CDRZ$	$GR = GR_{ctr} \wedge GF_{dat} \wedge GF_{res} \wedge GF_{req}$

Описанные в таблице стратегии можно разделить на чистые и комбинированные. К чистым можно отнести стратегии, которые используют только один механизм проверки готовности акторов к выполнению – это C -, D -, Z - и R -стратегии. Комбинированные стратегии используют некоторую комбинацию стратегий. Здесь возможно, что для определения готовности каждого актора проверяются несколько условий, либо отдельные готовности проверяются применительно к отдельным операторам. К комбинированным можно отнести следующие стратегии: ZR , DZ , DR , DZR , CZ , CZR , CD , CDZ , CDR . Особое место занимают вырожденная nil -стратегия и избыточная $CDRZ$ -стратегия.

Вырожденная nil -стратегия не задает ни одного механизма проверки готовности акторов. Эту ситуацию можно интерпретировать различным образом. Можно считать, что сигнал готовности формируется внутри самого актора, например по таймеру. В другой интерпретации ее можно рассматривать, как непрерывный во времени запуск, как например, в аналоговых системах.

Частные модели. Абстрактные машины. Абстрактная машина реализации стратегий управления. Стратегии управления БП можно описать в терминах виртуальной машины – машины реализаций стратегий управления БП, структура которой показана на рисунке.



Машина включает 3 подсистемы: репозиторий акторов, пул ресурсов и подсистему проверки готовности и запуска акторов.

Репозиторий акторов хранит произвольное число акторов (операторов). Каждый актор может быть либо собственно актором, либо контейнером, в котором хранится произвольное число акторов. Глубина вложений не ограничена.

Будем считать, что акторы относятся к одному БП, а связи между акторами есть часть описания самих акторов.

Пул ресурсов определяет набор доступных ресурсов. Каждому ресурсу ставится в соответствие вектор состояния (свободен, занят).

Подсистема проверки готовности и запуска акторов определяет порядок запуска акторов.

Выполнение БП в терминах МРСУ рассматривается как процесс запуска акторов в соответствии с определенными правилами. Правила определяются используемой стратегией.

Чистые стратегии. В рамках развиваемого подхода выделяется 4 чистые стратегии.

Директивная – С-стратегия. Данная стратегия предполагает, что все акторы определены до начала выполнения БП и связаны между собой связями по управлению (Control Flow), связи по данным отсутствуют (проверяются на этапе программирования), ресурсы также назначаются в статике и на этапе выполнения считается, что все необходимые ресурсы определены и свободны. Параллелизм описывается операторами *ветвление* и *слияние*. Этот вариант можно рассматривать как чистый поток управления (Control Flow).

Потоковая D-стратегия. Данная стратегия предполагает, что все акторы определены до начала выполнения БП и связаны между собой только связями по данным. Ресурсы закрепляются на этапе программирования, директивное управление не используется. Этот вариант можно рассматривать как чистый Data Flow [7].

Ресурсная R-стратегия. Данная стратегия предполагает, что все акторы готовы к выполнению и нужно только назначить ресурсы для этого. В таком случае стратегия представляет собой набор бизнес-правил и используется, когда акторы никак не связаны между собой, например когда ОС назначает процессы на выполнение.

Запросная Z-стратегия. Данная стратегия предполагает, что все акторы готовы к выполнению, а порядок их выполнения определяется потребностью в результатах выполнения актора. Можно выделить два варианта: 1) все запускаемые акторы известны и правила определяют только порядок их выполнения; 2) акторы могут генерироваться в процессе выполнения БП. По такому принципу работали рекурсивные машины [8], аналогичный принцип используется в функциональных языках программирования, а также в системах, работающих по принципу Map Reduce.

Комбинированные стратегии. Чистые стратегии на практике используются редко, чаще применяется некоторая их комбинация. При этом следует различать два случая: 1) на разных уровнях

используются разные стратегии; 2) комбинация стратегий используется на одном уровне. Далее рассматривается второй случай, когда одновременно проверяются несколько условий готовности.

Запросно-ресурсная стратегия предполагает, что в качестве основной стратегии используется запросная стратегия, в которой вычисления начинаются с вызова какой-либо корневой программы, которая может, в свою очередь, вызывать вложенные программы. Объем доступных ресурсов при этом ограничен. Эта стратегия используется, например, при реализации программы Лисп на однопроцессорной системе.

Потоково-запросная (она же *запросно-потоковая*) стратегия предполагает, что процесс выполняется за 2 прохода. На первом проходе с использованием механизма запроса строится дерево вычислений, а когда дерево построено, программа выполняется в потоковом режиме. При этом считается, что объем ресурсов не ограничен. Такая ситуация наблюдается при использовании виртуальных ресурсов.

Потоково-запросно-ресурсная стратегия аналогична *запросно-потоковой*. Отличие состоит в том, что ресурсы ограничены – например, несколько операторов могут быть назначены на выполнение на один процессор.

Директивно-запросную стратегию можно определить как стратегию, в которой совмещены директивное и запросное управление. Примером может служить традиционное процедурное программирование. При этом считается, что объем ресурсов не ограничен. Такая ситуация наблюдается при использовании виртуальных ресурсов.

Директивно-запросно-ресурсная стратегия аналогична *директивно-запросной* стратегии, но объем доступных ресурсов ограничен.

Директивно-потоковая стратегия (она же *потоково-директивная*) предполагает совместное использование механизмов потокового и директивного управления. В частности, такая стратегия использовалась в машинных потоках данных [7] и реализовывалась в форме подтверждений.

Директивно-потоково-запросная стратегия предполагает совместное использование *запросной* и *потоковой* стратегий в условиях ограниченности ресурсов. Этот вариант может быть интересен, когда БП организуется на двух уровнях: на верхнем (на уровне крупных модулей) реализуется *запросная* стратегия, а на нижнем – *потоковая*.

Директивно-потоково-ресурсная стратегия – то же самое, что *директивно-потоковая*, но с ограничением ресурсов.

Подходы к реализации и решаемые задачи.

Следует заметить, что использование предлагаемой метамодели и моделей, основанных на ней, не предполагает отказ от заделов, имеющихся в области Process Mining. Речь идет, скорее, о способе формирования лог-файлов и построении на их основе более сложных моделей, которые могут быть полезны для решения задач мониторинга и управления сложными техническими системами с динамической структурой. К таким задачам, в частности, могут быть отнесены: 1) реализация алгоритмов реструктуризации БП; 2) оптимизация использования ресурсов; 3) мониторинг БП, использующих запросные стратегии управления.

Задача реструктуризации может возникнуть по нескольким причинам: изменение структуры аппаратно-программной платформы, на которой выполняется наблюдаемый БП (контекстная реструктуризация), реструктуризация БП с целью адаптации к поступающим данным (контентная реструктуризация), изменение вычислительной обстановки. Для решения этой задачи представляется целесообразным использование стратегий, основанных на восстановлении графа потока данных с последующим наложением на него ресурсного графа. Для этого может использоваться система бизнес-правил.

Задача оптимизации ресурсов обычно возникает в случаях, когда изменяется вычислительная

обстановка, например требуется сократить время отклика за счет увеличения объема используемых ресурсов или, наоборот, уменьшить объем используемых ресурсов за счет уменьшения точности. В этом случае на первый план выходит задача мониторинга ресурсов. По результатам мониторинга принимается решение о реструктуризации БП.

БП, использующие запросные стратегии управления. Запросные стратегии находят все большее применение в реальных распределенных системах. Использование подобных стратегий тесно связано с использованием функциональных языков программирования.

В результате проведенных исследований проанализированы требования, предъявляемые к моделям, которые могут быть положены в основу построения СМ, и предложено использовать модель, основанную на стратегиях вычислений. Предложено выделять 16 базовых стратегий вычислений, которые могут использоваться при построении бизнес-процессов. Разработанную модель предлагается использовать в качестве метамодели, на базе которой могут строиться частные модели. Предлагаемые модели позволяют решать ряд задач управления БП, которые не могут быть решены при использовании традиционных моделей, основанных на моделях потоков управления.

СПИСОК ЛИТЕРАТУРЫ

1. Fundamentals of business process management second edition / M. Dumas, M. La Rosa, J. Mendling, H. Reijers // URL: <https://www.springer.com/gp/book/9783662565087#reviews> (дата обращения 07.06.2020).
2. ITIL–IT service management. URL: <https://www.axelos.com/best-practice-solutions/itil> (дата обращения 09.06.2020).
3. Computational and technological models of cognitive monitoring systems / V. U. Osipov, A. I. Vodyaho, N. V. Klimov, N. A. Zhukova, M. A. Chervontsev // Advances in Science, Technology and Engineering Systems Journal. 2019. Vol. 2, № 1. P. 197–202.

4. Van der Aalst W. Process mining data science in action. 2nd ed. Heidelberg: Springer, 2016.
5. Agha G. A. Actors: A model of concurrent computation in distributed systems. URL: <http://hdl.handle.net/1721.1/6952> (дата обращения 12.06.2020).
6. Функционально-ориентированные процессы / А. И. Водяхо, В. Б. Смолов, В. У. Плюснин, Д. В. Пузанков // под ред. В. Б. Смолова. Л.: Машиностроение. Ленингр. отд., 1988.
7. Veen A. H. Dataflow machine architecture // ACM Computing Surveys. 1986. № 18 (4). P. 365–396.
8. Venner J. Pro Hadoop. CA: Apress, Berkeley, 2009.

S. A. Abbas

Saint Petersburg Electrotechnical University

BUSINESS PROCESS MODELS FOR MONITORING SYSTEMS

Modern information systems differ in their complexity, which is reflected in the increase in the number of elements and in the complexity of the structures of business processes. To determine the state of complex technical systems, various monitoring systems are used, in particular, using Process Mining algorithms. However, modern Process Mining systems are largely focused on simple business process models that use only control flows. To solve real problems, it is required, in addition to control flows, to build models that take into account data flows and resources. The paper proposes a new BP model based on calculation strategies and considers 16 different control strategies. Using the proposed approach allows us to solve a number of important tasks related to control business processes.

Business processes, monitoring systems, internet of things