Шумский Л.Д.

СЕМАНТИЧЕСКАЯ ТРАССИРОВКА ИНФОРМАЦИОННЫХ ПРОЦЕССОВ

Аннотация: *В настоящее время активно прорабатываются вопросы использования формальных средств моделирования для описания различных категорий процессов, в частности информационных бизнес-процессов. Однако, в основном, в качестве средств моделирования используются графовые или сетевые модели, основанные на диаграммах состояний – такие как сети Петри, графы, представляющие сетевые цепочки, а также различные документно-ориентированные или событийно-ориентированные модели, такие как UML или IDEFx модели. Цель данной работы заключается в том, чтобы показать, что наработки в области теории бизнес процессов, ориентированные на такие модели, могут быть применены для более строгих символьных моделей, дающих возможность использования автоматизированной обработки модели процесса для проверки корректности, обнаружения свойств и связь модели процесса с технологическими средствами реализации. В данной работе, предлагается использование символьного средства моделирования процессов – pi-исчисление. В данной формальной модели процесс представляется как терм исчисления, выполнение которого описывается редукцией данного терма в соответствии с выбранной семантикой. Данное исчисление было разработано для описания взаимодействия нескольких систем в рамках процессов с возможно изменяющейся структурой. В данной работе предлагается оригинальный конструктивный подход к описанию трассировки – предлагается способ построения логов процесса, объединения их в журналы выполнения, рассматриваются общие требования к журналированию процессов. Описывается применение аспектов process mining к процессам, моделируемым с использованием pi-исчисления.Использование предлагаемых подходов к моделированию процессов и трассировки их выполнения предоставляет, по сравнению с аналогами, гораздо больше возможностей оценки адекватности и корректности построенной модели, упрощает расширение системы оценки для добавления новых критериев, упрощает получение и интерпретацию логов процесса, соответствующего модели.*
Ключевые слова: *Трассировка процессов, Моделирование бизнес-процессов, Process mining, pi-исчисление, лямбда исчисление, ABC, Исчисление взаимодействующих систем, Семантика выполнения бизнес-процессов, Интерпретация формальной модели, Оценка модели*

Введение

Средства, используемые сейчас для моделирования и управления бизнес-процес-

DOI: 10.7256/2305-6061.2014.1.11362

сами, как правило, не используют формальное описание или используют лишь общую его схему. Однако, использование формальной теории может принести существенные преимущества системе, которая их реализует и которая с ними строго соотносится. К этим преимуществам относятся простота и большая поддерживаемость разработанной системы, возможность ее расширения для включения новой функциональности, без изменения логического «ядра», лучшая контролируемость, некоторая дополнительная автоматизированная верификация и многие другие.

Вопрос использования формальной модели для описания процессов прорабатывается в различных направлениях, однако, в основном, используются графовые или сетевые модели, основанные на диаграммах состояний – такие как сети Петри [1], графы, представляющие сетевые цепочки [7], а также различные документно-ориентированные или событийно-ориентированные модели, такие как UML или IDEFx модели. Каждое из этих средств моделирования обладает достаточно высокой степенью наглядности и выразительности, благодаря чему широко используется, когда моделированием процесса производится специалистами вручную. Если же возникает необходимость автоматизированной обработки модели процесса, то лучше подходят более формально строгие средства символьного моделирования процессов. В данной работе, предлагается использование такого символьного средства моделирования процессов – π-исчисление [11], разновидность исчисления взаимодействующих систем [8]. В данной формальной модели процесс представляется как терм исчисления, выполнение которого описывается редукцией данного терма в соответствии с выбранной семантикой. Данное исчисление было разработано для описания взаимодействия нескольких систем по процессам с изменяющейся структурой. В отличие от обычных сетевых моделей π-исчисление полностью устраняет грань между переменными, константами, каналами передачи данных и другими используемыми элементами. Все объекты исчисления представляются именами, а конкретный смысл данных имен определяется доменом интерпретации процесса и семантикой выполнения. Во-первых, благодаря такому подходу описанные процессы не привязаны ни к какой конкретной реализации, а представляют собой чистое описание структуры процесса. Реализуя эту структуру в различных предметных областях, мы можем получать ее означивание. Это позволяет использовать π-исчисление как промежуточную модель для нескольких различных средств моделирования. Подробнее, интерпретация и означивание процесса будут описаны в первом разделе. Во-вторых, использование такой модели позволяет описывать процессы гораздо более сложной структуры, чем графовые модели, т.к. позволяет описывать передачу и выполнение подпроцессов, динамическое построение структуры процесса, выполнение функций высших порядков, описанных с помощью λ-исчисления [6, 9] и использовать другие средства повышения выразительности модели.

Цель данной работы заключается в том, чтобы показать, что наработки в области теории бизнес процессов, ориентированные на графовые модели, могут быть применены для более строгих символьных моделей. Для этого мы свяжем основные понятия теории процессов с процессами, представленными в виде термов исчисления, а также проде-

DOI: 10.7256/2305-6061.2014.1.11362

монстрируем, как с помощью процессов π-исчисления могут быть решены основные задачи моделирования процессов. Установка такой связи позволит шире использовать методологию π-исчисления для моделирования бизнес-процессов. Мы остановимся на аспектах, связанных с направлением process mining. Это направление заключается в исследовании связи моделей и реальным выполнением процессов, определении степени соответствия между моделями и объектами моделирования и возможность его применения крайне важно к моделированию бизнес процессов, т.к. при выполнении процесса существенную роль играет человеческий фактор. Первым шагом будет определено, как может быть представлен журнал выполнения процесса, для процесса, выполнение которого описывается шагами применения правил абстрактной машины. На основании определения журнала выполнения процесса π-исчисления будет дано определение общей модели процесса в терминологии process mining и будут определены правилам проверки построенной модели реальными данными – вычислении соответствие, простота, точность и способность к обобщению. Отдельное внимание будет уделено правилам автоматизированной композиции и декомпозиции таких процессов.

Описание π-исчисления и семантики выполнения процессов

Алфавит π-исчислении состоит из следующих компонентов. Множество «имен» исчисления *N*, обозначаемых малыми латинскими буквами. Множество процессов, означаемых большими латинскими буквами. Термы π-исчисления строятся индуктивно, с помощью настройки над существующими процессами или объединения сложных процессов. Эти операции описываются следующей грамматикой:

$$P ::= \mathbf{0} \mid \bar{x}y.P \mid x(y).P \mid (x)P \mid (P|Q) \mid !P$$

В грамматике используются следующие обозначения – символом 0 обозначается пустой процесс, т.е. процесс не выполняющий никаких действий. Префикс $\bar{x}[y_1, \dots, y_n]$ [10] обозначает передачу имен $y_1, \dots, y_n$ по ссылке *x*, префикс $x[y_1, \dots, y_n]$означает получение данных. Префикс получения данных связывает получаемые имена в процессе-продолжении и получение данных заключается в замене всех вхождений получаемого имени на полученное значение – если $x(y).P$ процесс получающий данные, а процесс $\bar{x}z$ передает данные, то результатом их коммуникации будет процесс $P[z/y]$. Коммуникация (прием-передача) осуществляется, когда по одному и тому же имени один процесс передает, а второй процесс получает данные. Кроме префикса получения данных существует возможность связать имя в процессе с помощью создания локальной ссылки $(x)P$. Такая запись связывает имя *x* в процессе *P* – такое имя оказывается локально определенном в процессе, никакой внешний процесс не сможет взаимодействовать, используя это имя. Такая инструкция используется для определения внутренних, защищенных или временных каналов обмена данными. В π- исчислении определяется один способ записи взаимодействия процессов – их параллельное выполнение. Параллельно выполняемые процессы могут взаимодействовать по определенным правилам. Репликация процессов $!P$ означает процесс, для которого всегда можно получить новую работающую копию.

DOI: 10.7256/2305-6061.2014.1.11362

Примером такого процесса может служить передача данных из системы, когда система по запросу может предоставить набор данных, связанный с именами $y_1, \dots, y_n$ по каналу *x*, то это может быть записано процессом $!\left(\bar{x}(y_1, \dots, y_n)\right)$. Запись данного процесса без репликации будет означать однократную передачу, которая не может быть повторена в рамках данного процесса – разница между передачей сообщения-запроса и передачей хранящихся справочных данных. Множество имен $\mathcal{N}$ является максимально общим множеством идентификаторов в используемой теории. Из него могут быть выделены подмножества, в соответствии с решаемой задачей. Например, при использовании функций, описанных термами λ-исчисления, в процессах из множества имен выделяется подмножество переменных функций

Необходимо обратить внимание на то, что имена, используемые в записи процесса, являются абстрактными символами, им не соответствует никакие реальные объекты предметной области. Далее, мы опишем механизм интерпретации термов π-исчисления в предметной области, для связи термов исчисления с конкретными моделируемыми процессами и их характеристиками.

По определению, под α преобразованием понимается переименование всех вхождений связанной переменной в терме. В общем виде, если $o[x_1, \dots\ x_n]$ это префикс, который связывает в терме переменные $x_1 \dots x_n$, то подстановка $o[y_1/x_1, \dots, y_n/x_n].\,[y_1/x_1, \dots, y_n/x_n]P$ называется α преобразованием. Если терм Q может быть получен из терма P путем конечного числа α преобразований, то такие термы называются α эквивалентными. В π-исчислении существует два способа связать переменную. Во-первых, префикс получения данных связывает получаемые имена, а, во-вторых, создание локальной переменной связывает ее в терме, в котором она создается.

Для описания выполнения процесса мы будем использовать Химическую Абстрактную Машину [5]. Для выполнения процесса в терминах данной модели процесс представляется в виде «молекулярного решения» или просто «решения» к которому применяются правила «реакции». Реакции в решении могут иметь как обратимый характер, обозначаемый символом ↔ , или необратимый, обозначаемый символом → . Химическая абстрактная машина будет использоваться, т.к. она обладает большей гибкостью и проще расширяема, чем простая операционная семантика. Для описания простейшей функциональности π-исчисления будет использоваться следующий набор правил:

$\{\|"P \mid Q \ \|\}" \leftrightarrow "\{\|"P\ Q\ \|\}"\ "$	Параллельное выполнение
$\{\|!P\ \} \leftrightarrow \{\|P\ !P\ \}$	Репликация
$\{\|(x\ P, Q\ 1, \dots, Q\ n \mid\} \rightarrow (x\ \{\|P\ Q\ 1, \dots, Q\ n \mid\}, x \quad F\ N\ Q\ 1, \dots, Q\ n)$	Локальная переменная
$\{\|\mathbf{0}\ P\ \} \leftrightarrow \{\|P\ \}$	Пустой процесс
$\{\|\bar{x}y\ P\ \} \rightarrow \{\|\bar{x}y\ P\ \}$	Асинхронное взаимодействие
$\{\|\bar{x}y\ x\ z\ .Q\ R\ \} \rightarrow \{\|[y \quad z\ \|Q\ R$	Коммуникация

Для описания механизма интерпретации термов исчиления предметной области введем следующие определения. Выделим из множества доступных имен π-исчисления $\mathcal{N}$ подмножество $\mathcal{L} \in \mathcal{N}$ попарно различных переменных. Будем называть множество

DOI: 10.7256/2305-6061.2014.1.11362

$\mathcal{L} \cup \tau$ множеством меток, меткой может быть или имя π-исчисления или специальный флаг τ, обозначающий отсутствие известной метки. Отличие меток от обычных имен π-исчисления заключается в отношении меток со связыванием переменных и α преобразованием. Особенностью меток является то, что метка не может быть связана и не может подвергаться α преобразованию. С добавлением меток грамматика построения термов изменяется следующим образом:

$P ::= \bar{x}[y_1, \ldots, y_n].P, \{x, y_1 \ \ldots, y_n\} \subseteq \mathcal{N}$ $\quad P ::= x(y_1, \ldots, y_n).P, x \in \mathcal{N}, \{y_1, \ldots, y_n\} \subseteq \mathcal{N} \backslash \mathcal{L}$

$P ::= (x)P, x \in \mathcal{N} \backslash \mathcal{L}$ $\quad P ::= P|Q$ $\quad P ::= !\,P$

Будем обозначать данную грамматику Γ_l. Если некоторый терм процесса P выводим в данной грамматике, то запишем это как $\Gamma_l \vdash P$. Как видно из определения грамматики, метками могут обозначаться либо передаваемые данные, либо имена-каналы, по которым происходит передача-получение данных.

Как уже было сказано раньше, обычный терм π-исчисления описывает структуру процесса, а не сам процесс предметной области. Разница между описанием структуры процесса и процессом предметной области заключается в понимании редукции соответствующих термов. Редукция терма чистого π-исчисления не несет никакого дополнительного смысла. С другой стороны, редукция процесса описывает выполнение процесса в некоторой предметной области – описывается использование определенных сущностей этой области и каждый зафиксированный шаг редукции процесса несет дополнительный смысл изменения состояния этих сущностей.

Будет использован ранее разработанный подход представления объектов предметной области [14], заключающийся в описании объектов как типизированных термов λ-исчисления, т.к. он позволяет хранить объекты предметной области в широком смысле – как, собственно, объекты данных (индивидные объекты, их атрибуты или отношения), так и вычислительные объекты – функции, определенные в этой области. Т.е. благодаря этому подходу, мы можем описать как обрабатываемые в процессе данные, так и правила, прописываемые в именах-каналах, которые описывают коммуникацию процесса с внешними системами. Смысл данного контекста заключается в связи абстрактных имен π-исчисления с используемыми объектами без нарушения структуры, определенной исчислением и семантикой. Терм π-исчисления, у которого контекст исполнения пуст или не содержит всех меток, определенных в терме, будем называть структурной моделью процесса M_S. Терм, контекст которого содержит значения для всех меток, будем называть концептуальной моделью процесса M_C.

$P, \Delta \in M_C \Rightarrow \Gamma_l \vdash P \ \& \ \forall n \in FN(P) \cap \mathcal{L} \rightarrow \ n \in \Delta$ Концептуальная модель пресса

$P, \Delta \in M_S \Rightarrow \Gamma_l \vdash P \ \& \ \exists n \in FN(P) \cap \mathcal{L} \rightarrow \ n \notin \Delta$ Структурная модель процесса

Разница между двумя этими типами моделей заключается в том, что нередексные термы в структурной модели исчисления могут быть редуцированы в концептуальной модели за счет выполнения функций, связанных с метками, из контекста выполнения процесса, для чего добавляются соответствующие правила к абстрактной машине выполнения. Абстрактная машина должна учитывать, что с выполняемым процессом связан контекст и, если стандартные правила не применимы, то модель может осуществлять

DOI: 10.7256/2305-6061.2014.1.11362

взаимодействие с помощью объектов концептуальной модели.

$\{|(l_1(y).P, R)(l_1, c_1; y, c_2; \Delta)|\} \to \{|([l_2/y]P, R)(l_2, \downarrow_\beta c_1 c_2; \Delta)|\}$ Получение данных
$\{|(\overline{l_1} l_2, R)(l_1, c_1; l_2, c_2; \Delta)|\} \to \{|(\downarrow_\beta c_1 c_2, R)(\Delta)|\}$ Передача данных

Правило внешнего получения данных описывает получение данных по каналу, описываемому меткой – в процессе продолжении связанная переменная изменяется на новую метку, которая раньше не была описана в этом процессе, а к контексту выполнения добавляется значения для этой метки равное редукции аппликации связанной с меткой-каналом функции и сущности связанной переменной. Последняя играет роль контейнера или заготовки для получения данных из внешней системы. Передача данных с помощью метки описывается как выполнение в рамках редукции процесса функции, связанной с меткой, аргументами которой служат объекты концептуальной модели, связанные с данными, данные, передаваемыми по этому каналу.

Таким образом, сохраняя полное и выполнимое описание структуры выполняемого процесса, мы получаем возможность моделировать обмен данными между процессом и внешним миром, выполнение и расчет предусловий процесса, а также описание структуры и наполнения передаваемых данных. Если терм $P \in M_C$ и $Q \in M_S$ и $P =_\alpha Q$, то мы будем говорить, что процесс *P* реализует структуру процесса *Q*, или, короче, что процесс *P* является конкретной реализацией процесса *Q*.

Под интерпретацией процесса π-исчисления мы будем понимать функцию, которая ставит в соответствие структуре процесса его конкретную реализацию в некоторой предметной области.

$$\cdot^{\mathcal{J}}: M_S \times \Delta^{\mathcal{J}} \to M_C; \Gamma_l \vdash P \in M_S \to \Gamma_l \vdash P^{\mathcal{J}} \in M_C$$

Применение функции интерпретации к структурной модели процесса, с использованием множества $\Delta^{\mathcal{J}}$,

называемому доменом интерпретации, как контекста, в котором выполняется вычисление, в результате дает концептуальную модель этого же процесса. Домен интерпретации представляет собой словарь, аналогичный контексту выполнения процесс, заполненный объектами предметной области той предметной области, в которой мы интерпретируем процесс. Идея интерпретации процесса заключается в последовательном наполнении контекста процесса и замене меток, которые не могут быть означены в этой предметной области на неиспользуемые в записи терма новые имена.

$$(\mathbf{0}, \Delta)^{\mathcal{J}} = \mathbf{0}, \Delta$$
$$(P \mid Q, \Delta)^{\mathcal{J}} = P^{\mathcal{J}} \mid Q^{\mathcal{J}}, \Delta$$
$$(!\, P, \Delta)^{\mathcal{J}} = !\, P^{\mathcal{J}}, \Delta$$
$$(\bar{x}[y_1, \dots, y_n].P,\ \Delta)^{\mathcal{J}} = \overline{[x, \alpha]_{\mathcal{J}}}[[y_1, \alpha]_{\mathcal{J}}, \dots, [y_n, \alpha]_{\mathcal{J}}].P^{\mathcal{J}}, \Delta \cup \Delta^{\mathcal{J}}(x, y_1, \dots, y_n)$$
$$(x(y_1, \dots, y_n).P)^{\mathcal{J}} = [x, \alpha]_{\mathcal{J}}(y_1, \dots, y_n).P^{\mathcal{J}}, \Delta \cup \Delta^{\mathcal{J}}(x)$$

Оператор $[\cdot, \alpha]_{\mathcal{J}}$ возвращает свой первый аргумент, если он присутствует как ключ в домене интерпретации, и альфа преобразование этого аргумента, относительно терма, в котором он используется, в противном случае.

DOI: 10.7256/2305-6061.2014.1.11362

Аналогичный подход к интерпретации термов формального языка в конкретной предметной области используется в области семантического моделирования предметных областей, например в дескрипционной логике [4]. Использование такого алгоритма для интерпретации модели процессов позволяет использовать крайне гибкую общую схему. Согласно этой схеме, модель процессов и объектов, обрабатываемых в нем, интерпретируются сходным образом и относятся в каждой предметной области к одним сущностям. Это дает возможность использовать разработанные модели, как шаблоны, в любой предметной области без изменения самой модели – достаточно определить для нее корректный домен интерпретации.

Для концептуальной модели процесса необходимо расширить понятие бисимулярности (слабого равенства) процессов. В классической версии **π-исчисления под бисимилярностью** процессов P и Q $-$ $P \cong Q$ понимается следующее отношение, считая что $\mathbf{0} \cong \mathbf{0}$.

$$P \cong Q \Leftrightarrow P \downarrow_{\alpha} = P' \rightarrow Q \downarrow_{\alpha} = Q' \& P' \cong Q'$$

Т.е. если два процесса бисимулярны, то если один из них может быть редуцирован по префиксу *α*, то и второй может быть редуцирован по тому же префиксу и результаты редукции также будут бисимулярны. Связь с объектами канонической модели добавляет ограничение на то, что метки процессов должны быть связаны с эквивалентными объектами концептуальной модели. Кроме того, если процесс может быть редуцирован с помощью правила коммуникации с внешними системами, то метки участвующие в такой коммуникации также должны совпадать.

Основные определения

Основное назначение любой модели – адекватно с некоторой определенной точки зрения отражать реальные объекты предметной области и выдвигать гипотезы об объектах предметной области, которые могут быть формально проверены. Исходя из этой установки, для успешного применения любого средства моделирования желательно определить правила (оценивающие функции), позволяющие проверять корректность модели. Для моделирования процессов в качестве таких правил используются оценки соответствия, простоты, простота и способность к обобщению [2]. Данный раздел посвящен определению и описанию данных оценок к моделям π-исчисления.

Отправной точкой для любых оценок корректности модели процесса является понятие лога процесса. В общем случае, журнал (лог) представляет собой множество записей содержащих следы функционирования отдельных процессов, записанных в терминах выполненных действий. Анализ применимости модели для описания конкретного процесса основывается на том, как модель соотносится с существующими журналами – любой ли журнал может быть получен из модели, какое множество новых записей журнала может быть получено и т.д.

Сетевые и графовые средства моделирования процессов используют для описания журнала выполнения процесса теоретико-множественные понятия, такие как множества

DOI: 10.7256/2305-6061.2014.1.11362

и операции над ними, последовательности, проекции и прочее. Это оправдано, т.к. эти понятия хорошо соотносятся с графовыми моделями. Для π-исчисления мы воспользуемся подходом, используемым в аппликативных вычислительных системах (АВС), согласно которому все объекты модели представляются термами, построенными по одному индуктивному алгоритму, а различие между ними заключается только в их взаимодействии с другими объектами модели [15].

- Объектами АВС являются сущности, арность которых не фиксируется заранее, а проявляется постепенно, по мере взаимодействия объектов;
- Конструирование новых составных объектов выполняется с помощью операции аппликации (применения) одних объектов к другим, что может трактоваться как применение функции к ее аргументу в классической теории, однако является более широким понятием и может трактоваться соответствующим образом, в зависимости от трактовки самих объектов. Необходимо отметит, что в АВС отсутствует понятие изменяемого объекта. Аппликация термов порождает новый терм, никак не изменяя старых;
- Роли функции и аргумента являются относительными. В различных случаях объект может выступать как в роли функции, так и в роли аргумента. Допускается, в частности, самоприменимость объектов.

Данный подход будет использоваться, т.к. в π-исчислении нет отдельно выделенного понятия «действие», на которое опирается теоретико-множественное определение записи журнала. Внутренняя структура модели, представленной термом π-исчисления не известна до того, как процесс начнет выполняться. Выполнение такой модели дает не обход графа, который может быть представлен множеством, а последовательность редексов. Причем из-за особенности абстрактной машины, с помощью которой выполняется редукция термов, заключаещей в недетерминированном выборе пары взаимодействующих элементов, редукция одной модели на одинаковых входных данных может вернуть различные последовательности редексов.

Чтобы учесть эти особенности используемого средства моделирования, журнал или запись журнала должны представлять собой терм исчисления, удовлетворяющий следующим условиям:

1. Журнал выполнения может быть построен только по концептуальной модели процесса. Журнал выполнения структурной модели процесса не имеет смысла, т.к. редукция такого терма не зависит ни от чего, кроме самого терма и выбранной стратегии редукции;
2. Журнал должен отслеживать состояния терма в терминах, существующих в π-исчислении – редукции терма, редексов процесса и т.д.;
3. Журнал должен быть вычислимым объектом. Для журнала должно быть определено правило совместного выполнения его и модели. Такое совместное выполнение позволит оценить соответствие модели процесса журналу, правила такого выполнения будут описаны далее. Пока необходимо сказать, благодаря тому, что журнал является вычислимым объектом, для проверки соответствия модели нет необходимости использовать внешние функции. Вся необходимая информация содержится в записях журнала выполнения. Такая особенность, в частности, позволяет

DOI: 10.7256/2305-6061.2014.1.11362

динамически определять подходящую модель для журнала, без необходимости описания всех возможных функций проверки.

Журнал выполнения концептуальной модели процесса будет представляться списком объектов, представляющих одну запись журнала. Объект записи представляет собой некоторую функцию предопределенной структуры, относящуюся к определенным характеристикам выполнения терма химической абстрактной машиной. Далее в работе мы будем использовать следующие характеристики – тип выполняемого действия (отслеживаемые реакции химической абстрактной машины), задействованные метки (количество которых определяется типом реакции) и объекты канонической модели, связанные с данными метками. Для определения и выделения данных характеристик в ходе выполнения процесса, воспользуемся расширением используемой абстрактной машины, которое будет отслеживать выполнение редукции. Идея вводимого расширения заключается в том, что правила реакции химической абстрактной машины не позволяют восстановить ход выполнения процесса, поэтому нам необходимо выполнять реакцию с дополнительным оператором обзора. Обозначим такой оператор *N* и представим для него следующие правила:

$$N(\{|(l_1(y).P,R)(l_1,c_1;y,c_2;\Delta)|\}) = [eo,(l,y),(c_1,c_2)] :: N\left(\left\{\left|([l_2/y]P,R)\left(l_2,\downarrow_\beta c_1c_2;\Delta\right)\right|\right\}\right)$$
$$N\left(\left\{\left|\left(\overline{l_1}l_2,R\right)(l_1,c_1;l_2,c_2;\Delta)\right|\right\}\right) = [ei,(l_1,l_2),(c_1,c_2)] :: N\left(\left\{\left|\left(\downarrow_\beta c_1c_2,R\right)(\Delta)\right|\right\}\right)$$
$$N(\{|(\bar{x}l_1,x(z).Q,R)(l_1,c_1;\Delta)|\}) = [c,(\tau,l_1),c_1] :: N\left(\left\{\left|\left[{}^{y}/_{z}\right]\right|Q,R\right\}\right)$$
$$N(S) = \boldsymbol{if}\ S \downarrow_{ChAM} = S\ \boldsymbol{then}\ S\ \boldsymbol{else}\ N(S \downarrow_{ChAM})$$

Запись $a :: b$ означает построение списка из двух элементов. По отношению к спискам мы будем использовать следующие стандартные обозначения – *Nil* : пустой список, *length*: функция, возвращающая длину списка, *head, tail*: функции возвращающие первый элемент списка и список без первого элемента, соответственно, *map*: функция, возвращающая список, составленный из результатов применения переданной функции к элементам исходного списка.

Сам оператор *N*, оператор контроля выполнения процесса, может присутствовать в любом варианте представления модели процессов, не зависимо от того, как в ней представляются различные аспекты моделирования. Данный оператор должен получать на вход молекулярное решение химической абстрактной машины и возвращать список отслеживаемых реакций. Последним элементом данного списка должен быть результат выполнения процесса – решение, в котором невозможно выполнить ни одной реакции. Такое поведение оператора обеспечивается четвертым правилом. Остальные правила, имеющие более высокий приоритет, описывают выделение характеристик выполнения процесса из отслеживаемых реакций. В данной работе мы будем рассматривать три типа реакций – обмен данными с внешними системами и коммуникация внутри процесса, которая затрагивает объекты концептуальной модели. Необходимо отметить, что имена π-исчисления, которые не являются метками не попадают в журнал, вместо них в журнал записывается пустая метка τ. Это связано с тем, что имя, не являющееся меткой, не является идентификатором и не имеет смысла после выполнения процесса, а значит в журнале оно будет только вводить в заблуждение.

DOI: 10.7256/2305-6061.2014.1.11362

Правила химической абстрактной машины неприоритезированы и выполнение их может быть в произвольном порядке. Оператор *N* позволяет разрешить эту проблему. Для его операций приоритет введен и общая приоритезация выполнения процесса осуществляется следующим образом. Если несколько правил абстрактной машины могут быть применены одновременно, то выбирается правило оператора *N* с наивысшим приоритетом и применяется. Таким образом, не изменяются правила выполнения самой абстрактной машины, но обеспечивается зависимость записи журнала только от самой модели процесса.

Запись журнала выполнения процесса будет представляться термом, отвечающем следующим требованиям – во-первых, он должна содержать информацию о самом ходе выполнения процесса, а, во-вторых, он должен содержать алгоритм проверки соответствия. В данной работе мы будем использовать самый простой и очевидный вариант построения такого терма, для увеличения наглядности:

$$T = \lambda v.\,(\lambda xyz.\,[v.1 = x]\ \&[v.2 = y]\&[v.3 = z])[t, l, c]$$

В данной записи константы t,l,c соответствуют конкретным значениям, относящимся к этой записи, а внутренне выражение выполняет простейшую проверку на соответствие между этой записью и параметрами, с которыми происходит сравнение. Необходимо отметить, что эта проверка, при сохранении типа, может иметь любую, сколь угодно сложную структуру, например учитывать проставленные веса меток и объектов канонической модели [3] или иные методы сравнения. Т.к. данная проверка входит в состав записи журнала выполнения процесса, то такой подход позволяет использовать наиболее подходящий алгоритм сравнения для каждой модели процесса и даже для различных подпроцессов одного процесса. Как уже говорилось выше, журнал, в простейшем случае, представляет собой список записей: $L = T_1 :: T_2 :: \cdots :: T_n$. Однако, при необходимости, данное понятие может быть расширено, чтобы описывать модальные характеристики наблюдаемых записей.

Для того, чтобы иметь возможность восстановить фрагмент процесса, который должен присутствовать в модели, согласно сравниваемому с ней журналу, введем оператор восстановления процесса *B*. Смысл данного оператора в том, чтобы по записи журнала получить процесс, редукция которого даст данную запись. Данный оператор требуется, если необходимо восстановить модель процесса по журналу его выполнения. В данной работе рассматривается слишком простой подход к созданию журнала, чтобы можно было полноценно восстановить модель, но на примере ее мы можем продемонстрировать подход к построению оператора *B*.

$$B(eo, (l_1, \dots, l_n), (c_1, \dots, c_n), S, \Delta) = \{\big|\big(l_1 l_2, \dots, l_n, S\big)(l_1, c_1; \dots; l_n, c_n; \Delta)\big|\}$$ Передача данных

$$B(ei, (l_1, \dots, l_n), (c_1, c_n), S, \Delta) = \{|(l_1(l_2, \dots, l_n).S)(l_1, c_1; \dots; l_n, c_n; \Delta)|\}$$ Получение данных

Использование оператора восстановления процесса B позволяет использовать более сложную логику проверки соответствия – она заключается в получении модели, полностью описывающей все записи журнала и проверки бисимулярности ее с исходной моделью. При простой проверке соответствия этот оператор необходим, для того, чтобы

DOI: 10.7256/2305-6061.2014.1.11362

проверять модель, если какая-то из записей журнала не может быть выведена из нее.

Простая функция проверки соответствия *Ch*, которая будет описана далее, должна принимать на вход журнал выполнения процесса и модель, соответствие с которой необходимо проверить. Результатом ее выполнения должна быть бинарная или непрерывная характеристика, показывающая степень соответствия процесса и модели. Т.к. вся логика проверки соответствия записи журнала и модели заложена в самом объекте записи, задача функции проверки соответствия заключается в том, чтобы обеспечить параллельное выполнение процесса с помощью оператора контроля выполнения и проход по журналу с применением объекта текущей записи к текущему шагу выполнения. После такого выполнения функция проверки должна корректно агрегировать результаты прохода и вернуть результат в соответствующем формате.

Функцию, выполняющую поставленную задачу мы будем строить поэтапно. Первым шагом является конструирование функции параллельного выполнения журнала и модели процесса. Задача параллельного выполнения журнала и процесса усложняется тем, что не на каждый шаг редукции процесса найдется запись в журнале из-за того, что некоторые шаги не записываются. Для того, чтобы решить эту проблему, определим оператор $\mathbb{N}$, который применяет к процессу оператор $\mathbb{N}$ до тех пор, пока не выполнит шаг, записываемый в журнал:

$$\mathbb{N}S = \boldsymbol{if}\ length(NS) = 1\ \boldsymbol{then}\ \mathbb{N}S\ \boldsymbol{else}\ NS$$

С учетом такого оператора, функция, которая выполняет параллельное выполнение журнала и процесса записывается следующим образом:

$$ChS\ L,P] = \boldsymbol{if}\ head(L)(head\ \mathbb{N}P)\ \boldsymbol{then}\ ChS(tail\ L, tail\ \mathbb{N}P)\ \boldsymbol{else}\ ChS\left(L, P \cup \left(B(head\ \mathbb{N}P)\right)\right)$$

Логика данной проверки заключается в том, что если некоторый элемент журнала не может быть получен из модели, то проверка продолжается с добавлением элемента, который строго соответствует недостающей записи.

Функция параллельного выполнения *ChS* ложится в основу построения простой функции проверки соответствия *Ch*. Сама она может быть построена, следуя одной из трех основных стратегий или их комбинаций:

1. Завершаться при получении первой записи, которая не может быть получена из модели;
2. Проводить проверку до конца, подсчитывая количество невыводимых записей и возвращать данное значение как результат;
3. Учитывать количество записей журнала, которые остались, после завершения выполнения процесса.

Функция проверки также может быть усложнена для получения корректной модели или ее точность может быть увеличена при использовании типизированного [12]. варианта исчисления. Для такого использования мы будем оперировать системой, в которой тип, приписанный имени, может быть получен с помощью оператора $\Delta(\_)$. Для меток – результатом выполнения этого оператора будет тип, содержащийся в контексте выполнения процесса, для простых имен – локальный тип, определенный в процессе.

DOI: 10.7256/2305-6061.2014.1.11362

Заключение

В данной работе предлагается подход к моделированию журналирования выполнения процесса с использованием формальной системы π-исчисления. Использование предлагаемого подхода к моделированию процессов и трассировки их выполнения предоставляет, по сравнению с аналогами, гораздо больше возможностей оценки адекватности и корректности построенной модели, упрощает расширение системы оценки для добавления новых критериев, упрощает получение и интерпретацию логов процесса, соответствующего модели. Реализация такого подход к журналированию была применена к распределенной вычислительной сети [13]. Был произведен анализ логов вычислительных процессов в такой сети, результаты которого позволили установить корректность применяемой модели вычисления и оптимизировать алгоритм распределения вычислительного процесса по узлам вычислительной сети.

Библиография :

1. Aalst W.M. van der. Formalization and verification of event-driven process chains // In-formation and Software technology. 1999. Т. 41. № 10. С. 639–650.
2. Aalst W.M. van der. Discovery, Conformance and Enhancement of Business Processes. : Springer, 2011.
3. Aalst W.M. van der, Adriansyah A., Dongen B.F. van. Conformance checking using cost-based fitness analysis // Enterprise Distributed Object Computing Conference (EDOC), 2011 15th IEEE International. , 2011. С. 55–64.
4. Baader F. и др. The description logic handbook: Theory, implementation and applications. : Cambridge university press, 2003.
5. Berry G., Boudol
6. Boudol G. The π-calculus in direct style // Higher-Order and Symbolic Computation. 1998. Т. 11. № 2. С. 177–208.
7. Jianzhong L., Liangyou C. Extended Event-Process Chain (EEPC) and It's Application in BPR // Systems Engineering. 2000. Т. 1. С. 009.
8. Milner R. A calculus of communicating systems. : Springer-Verlag New York, Inc., 1982
9. Milner R. Functions as processes // Mathematical structures in computer science. 1992. Т. 2. № 02. С. 119–141.
10. Milner R. The polyadic π-calculus: a tutorial. : Springer, 1993.
11. Milner R., Parrow J., Walker D. A calculus of mobile processes, i // Information and com-putation. 1992. Т. 100. № 1. С. 1–40.
12. Pierce B., Sangiorgi D. Typing and subtyping for mobile processes // Logic in Computer Science, 1993. LICS'93., Proceedings of Eighth Annual IEEE Symposium on. , 1993. С. 376–385.
13 humsky L. и др. Applicative Approach to Information Processes Modeling-Towards a Constructive Information Theory: : SciTePress-Science and and Technology Publications, 2013a. С. 323–328.
14 Shumsky L. и др. A synthetic approach to building a canonical model of subject areas in the integration bus // ISKO-Maghreb, 2013 3rd International Symposium. , 2013b. С. 1–7.

DOI: 10.7256/2305-6061.2014.1.11362

15. Вольфенгаген В.Э. Методы и средства вычислений с объектами. Аппликативные вы-числительные системы. : ООО «ЮрИнфоР-Пресс», 2004

References:

1. Aalst W.M. van der. Formalization and verification of event-driven process chains // In-formation and Software technology. 1999. T. 41. № 10. S. 639–650.
2. Aalst W.M. van der. Discovery, Conformance and Enhancement of Business Processes. : Springer, 2011.
3. Aalst W.M. van der, Adriansyah A., Dongen B.F. van. Conformance checking using cost-based fitness analysis // Enterprise Distributed Object Computing Conference (EDOC), 2011 15th IEEE International. , 2011. S. 55–64.
4. Baader F. i dr. The description logic handbook: Theory, implementation and applications. : Cambridge university press, 2003.
5. Berry G., Boudol G. The chemical abstract machine // Theoretical computer science. 1992. T. 96. № 1. S. 217–248.
6. Boudol G. The π-calculus in direct style // Higher-Order and Symbolic Computation. 1998. T. 11. № 2. S. 177–208.
7. Jianzhong L., Liangyou C. Extended Event-Process Chain (EEPC) and It's Application in BPR // Systems Engineering. 2000. T. 1. S. 009.
8. Milner R. A calculus of communicating systems. : Springer-Verlag New York, Inc., 1982.
9. Milner R. Functions as processes // Mathematical structures in computer science. 1992. T. 2. № 02. S. 119–141.
10. Milner R. The polyadic π-calculus: a tutorial. : Springer, 1993.
11. Milner R., Parrow J., Walker D. A calculus of mobile processes, i // Information and com-putation. 1992. T. 100. № 1. S. 1–40. 12.
12. Pierce B., Sangiorgi D. Typing and subtyping for mobile processes // Logic in Computer Science, 1993. LICS'93., Proceedings of Eighth Annual IEEE Symposium on. , 1993. S. 376–385.
13 Shumsky L. i dr. Applicative Approach to Information Processes Modeling-Towards a Constructive Information Theory: : SciTePress-Science and and Technology Publications, 2013a. S. 323–328.
14 Shumsky L. i dr. A synthetic approach to building a canonical model of subject areas in the integration bus // ISKO-Maghreb, 2013 3rd International Symposium. , 2013b. S. 1–7.
15. ol'fengagen V.E. Metody i sredstva vychislenii s ob"ektami. Applikativnye vy-chislitel'nye sistemy. : OOO «YurInfoR-Press», 2004.

DOI: 10.7256/2305-6061.2014.1.11362