УДК 004.5 DOI: 10.18698/0236-3933-2020-4-99-113

ВЕРИФИКАЦИЯ ЗНАНИЙ, ПОЛУЧЕННЫХ ПРИ ИЗУЧЕНИИ МОДЕЛЕЙ БИЗНЕС-ПРОЦЕССОВ

В.В. Девятков deviatkov@bmstu.ru
А.Р. Кадырбаева naten702@mail.ru

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Аннотация

Рассмотрена методика формальной верификации (проверки) знаний, полученных при изучении моделей бизнес-процессов. В качестве формального языка для представления и верификации бизнес-процессов выбран некоторый вариант π-исчисления, в качестве языка для формулировки тестовых заданий (вопросов), подлежащих проверке, — линейная временная модальная логика. Приведено обоснование такого выбора, рассмотрены принципы формирования тестовых заданий на языке модальной логики и их использования для верификации знаний. Автоматизацию проверки знаний предложено осуществлять логическими программами, получаемыми в результате перехода от описания бизнес-процессов к программе проверки правильности на языке логического программирования Visual Prolog. Методика формальной верификации знаний проиллюстрирована примерами проверки знаний. Показано, как от описания бизнес-процессов осуществляется переход к процедурам формальной верификации (проверки) знаний. Обсуждены перспективы развития предлагаемой методики

Ключевые слова

Верификация знаний, тестовые задания, модальная логика, язык логического программирования Пролог

Поступила 11.09.2019
Принята 02.10.2019

Работа поддержана грантом Минобрнауки России № 2.5048.2017/8.9

Введение. В настоящее время в связи с усложнением моделей и процессов, которые изучают будущие специалисты в области информационных систем и технологий, актуальны задачи формальной проверки уровня освоения ими знаний и автоматизации этой формальной проверки. Модели бизнес-процессов, используемые для проверки знаний, могут формироваться либо самими обучающимися по некоторому заданию, либо быть заранее сформированными. В том и другом случае для каждой по-

добной модели, используемой для проверки знаний, должен быть заготовлен список вопросов, на которые тестируемый должен ответить. По результатам ответов оценивается уровень знаний тестируемого.

Перечень представительного множества моделей бизнес-процессов, используемых для проверки знаний, не является предметом рассмотрения в настоящей работе. Предположим, что это множество моделей уже выбрано, а испытуемый строит одну из этих моделей на основании некоторого задания. Нашей задачей является только демонстрация принципов проверки знаний по этим моделям, причем после того, как модель построена испытуемым.

Поскольку бизнес-процессы — это динамические системы, то формальная проверка знаний бизнес-процессов может рассматриваться как специфическая область верификации динамических систем. К настоящему времени известно три основных подхода к верификации динамических систем: тестирование наличия у системы требуемых свойств, формальное доказательство наличия свойств в рамках какого-либо исчисления и моделеориентированное доказательство наличия свойств. Различие между этими подходами состоит в следующем.

Для проверки на вход модели бизнес-процесса подаются заранее подготовленные последовательности действий, представленные на каком-либо языке, и которые могут осуществляться при выполнении бизнес-процесса, а затем проверяется проявление моделью ожидаемых свойств. Важным недостатком такого тестирования является то, что мощность множества тестирующих последовательностей действий может быть слишком велика, что приводит к комбинаторному взрыву, а также вычислительным проблемам. Вместе с тем множества тестирующих последовательностей в большинстве случаев формируются обучающимися, что не дает никакой гарантии полноты представленного множества для проверки знаний.

В результате формальной проверки наличия свойств в рамках какого-либо исчисления требуемые свойства формулируются в виде целей или теорем на языке соответствующего исчисления, которые затем выводятся (доказываются) на основе начальных (предыдущих) знаний, представляющих собой модель бизнес-процесса на том же языке, используя при этом специальные правила вывода. Если стратегия проверки завершена, то успех доказательства наличия требуемых свойств зависит от полноты аксиоматизации начальных знаний. Кроме того, полная стратегия вывода может оказаться вычислительно сложной. Но главная проблема, ограничивающая использование исчислений, заключается в сложности интерпретации и аксиоматизации на языке принятых исчислений

моделей бизнес-процессов обучающими, которые, как правило, не являются специалистами в области исчислений.

В отличие от формального доказательства в рамках исчисления моделеориентированное доказательство наличия свойств использует непосредственно модели бизнес-процессов. Если создатель этих моделей считает, что он описал все предъявляемые к ним требования, то для проверки и валидации модели остается только доказать, что она удовлетворяет всем этим требованиям. Для подобного доказательства здесь опять же используются правила вывода, но семантика этих правил существенно ориентирована на привычные для обучающего понятия и смыслы.

Именно на моделеориентированном доказательстве свойств бизнес-процессов и основана настоящая работа. Одними из самых известных моделей, применяемых для таких целей, являются процессные модели. Прародителем целого класса таких моделей может выступать, например, язык процессных выражений процессной алгебры (π-исчисления) Роберта Милнера [1]. Известно, что выражения π-исчисления предлагают достаточно естественный путь описания бизнес-процессов [2], состоящих из бизнес-процессов, взаимодействующих по общим каналам. К сожалению, языки процессной алгебры изящны с математической точки зрения, но зачастую при их прямом использовании слишком абстрактны для применения.

В связи с этим популярным является использование так называемых языков архитектурного описания Architecture Description Language (ADL) [3], которые позволяют на разных уровнях абстракции описывать структуру и поведение бизнес-процессов. На сегодняшний день предложено довольно большое число различных языков архитектурного описания [4–7], рассчитанных на описание и проверку динамических систем. В настоящей работе мы пойдем именно этим путем, описывая архитектуру бизнес-процессов статически, а их поведение — простейшим вариантом процессной алгебры.

Далее в следующем разделе будут приведены основные понятия языка описания структуры и поведения бизнес-процессов. Затем на простом примере бизнес-процесса будет продемонстрировано использование языка. Применительно к этому же примеру будет показано, как формулируются на языке модальной логики свойства, которым должен удовлетворять бизнес-процесс, построенный обучаемым, чтобы считаться правильным. Далее изложены принципы верификации знаний, полученных обучаемым, использующие введенные понятия и смыслы. В заключении статьи обсуждаются направления дальнейших работ.

Описание структуры бизнес-процессов. В простейшем случае бизнес-процесс каждого уровня иерархии $i = 1, 2, \ldots, n$ будем изображать прямоугольником и представлять агентом A_n^i, где n — имя агента, являющегося четверкой $A_n^i = \left\{ A_n^{i+1},\ I_n,\ O_n,\ D_n \right\}$, где A_n^{i+1} — множество *агентов уровня* $(i + 1)$, *входящих в агент* A_n^i; I_n и O_n — множество внешних входных и выходных каналов агента A_n^i; D_n — множество внутренних каналов агента A_n^i.

Рассмотрим пример простого бизнес-процесса реализации заказа (БПРЗ), заимствованного из работы [8] и представленного в наших обозначениях. Бизнес-процесс верхнего 1-го уровня представлен агентом $A^1_{\text{БПРЗ}}$ на рис. 1 и состоит из четырех агентов 2-го уровня: покупателя, продавца, банка и производителя [9].

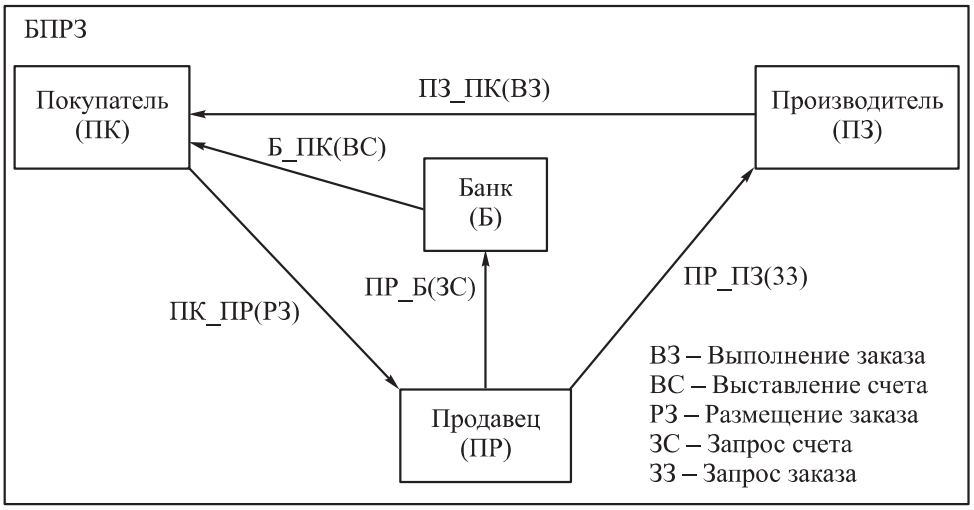

Рис. 1. Агент реализации заказа $A^1_{\text{БПРЗ}}$

Согласно приведенным обозначениям агент $A^1_{\text{БПРЗ}}$ имеет только внутренние каналы:

- $A^1_{\text{БПРЗ}} = \left\{ A^2_{\text{ПК}},\ A^2_{\text{ПР}},\ A^2_{\text{Б}},\ A^2_{\text{ПЗ}} \right\}$,
- $I_{\text{БПРЗ}} = \varnothing$,
- $O_{\text{БПРЗ}} = \varnothing$,
- $D_{\text{БПРЗ}} = \{\text{ПЗ\_ПК},\ \text{Б\_ПК}, \text{ПК\_ПР}, \text{ПР\_Б}, \text{ПР\_ПЗ}\}$.

Агенты 2-го уровня $A^2_{\text{ПК}}$, $A^2_{\text{ПР}}$, $A^2_{\text{Б}}$, $A^2_{\text{ПЗ}}$, входящие в $A^1_{\text{БПРЗ}}$, приведены на рис. 2–5.

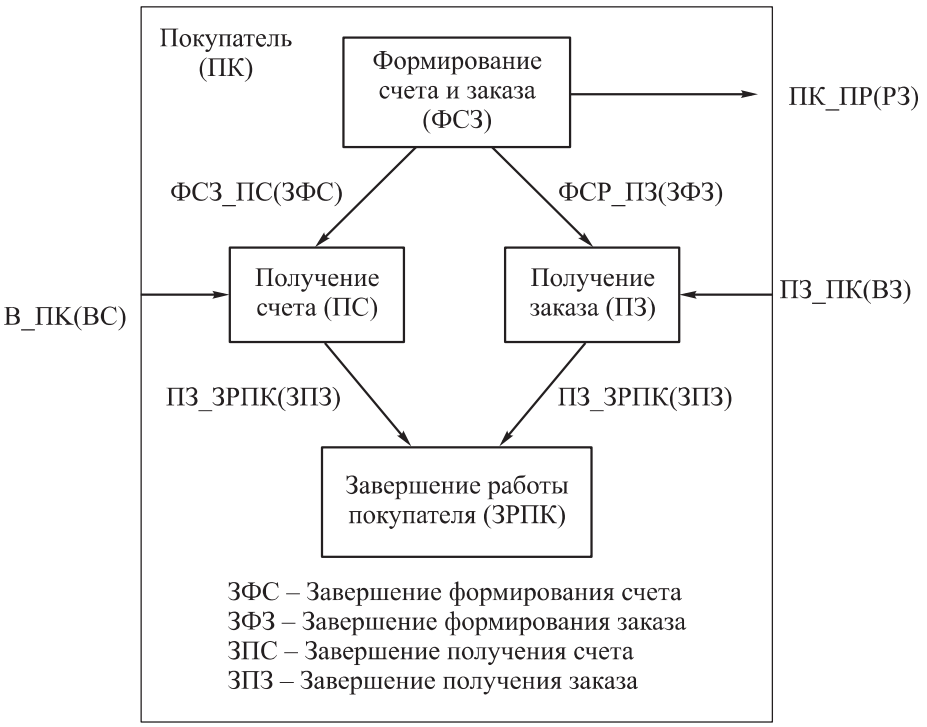

Рис. 2. Агент-покупатель $A^2_{\text{ПК}}$

Для агента $A^2_{\text{ПК}}$ (см. рис. 2) имеем:

- $A^2_{\text{ПК}} = \varnothing$,
- $I_{\text{ПК}} = \{\text{Б\_ПК}, \text{ПЗ\_ПК}\}$,
- $O_{\text{ПК}} = \{\text{ПК\_ПР}\}$,
- $D_{\text{ПК}} = \{\text{ФСЗ\_ПС}, \text{ФСР\_ПЗ}, \text{ПС\_ЗРПК}, \text{ПЗ\_ЗРПК}\}$.

Для агента $A^2_{\text{ПР}}$ (см. рис. 3) запишем:

- $A^2_{\text{ПР}} = \varnothing$,
- $I_{\text{ПК}} = \{\text{ПК\_ПР}\}$,
- $O_{\text{ПР}} = \{\text{ПР\_Б}, \text{ПР\_ПЗ}\}$,
- $D_{\text{ПР}} = \{\text{ФЗС\_ЗРПР}, \text{ФЗК\_ЗРПР}\}$.

Для агента $A^2_{\text{Б}}$, приведенного на рис. 4, имеем:

- $A^2_{\text{Б}} = \varnothing$,
- $I_{\text{Б}} = \{\text{ПР\_Б}\}$,
- $O_{\text{Б}} = \{\text{Б\_ПК}\}$,
- $D_{\text{ПР}} = \{\text{ФС\_ЗФС}\}$.

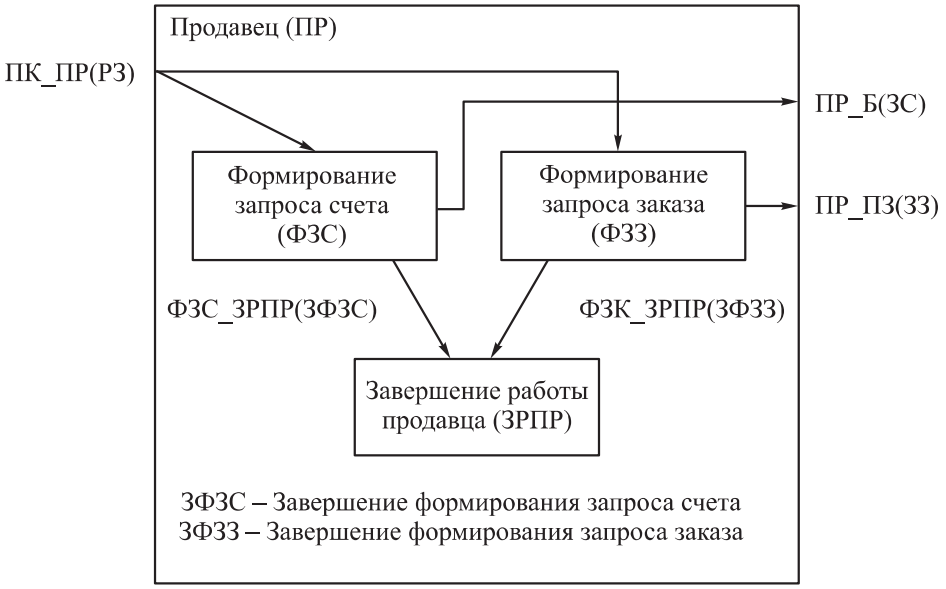

Рис. 3. Агент-продавец $A^2_{\text{ПР}}$

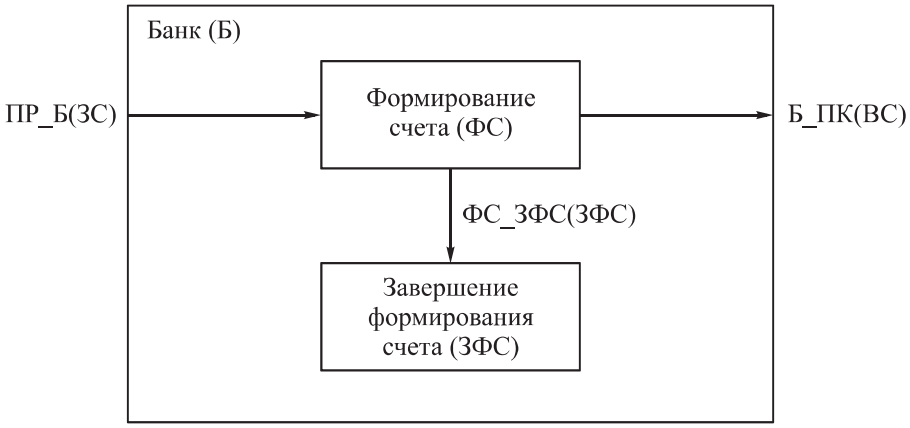

Рис. 4. Агент-банк $A^2_{\text{Б}}$

Для агента $A^2_{\text{ПЗ}}$ (см. рис. 5) запишем:

- $A^2_{\text{ПЗ}} = \varnothing$,
- $I_{\text{ПЗ}} = \{\text{ПР\_ПЗ}\}$,
- $O_{\text{ПЗ}} = \{\text{ПЗ\_ПК}\}$,
- $D_{\text{ПР}} = \{\text{ФВЗ\_ЗВЗ}\}$.

Описание поведения агентов. Приведенное описание структуры бизнес-процессов носит статический характер. Для описания поведения агентов будем использовать язык процессных выражений, заимствующий

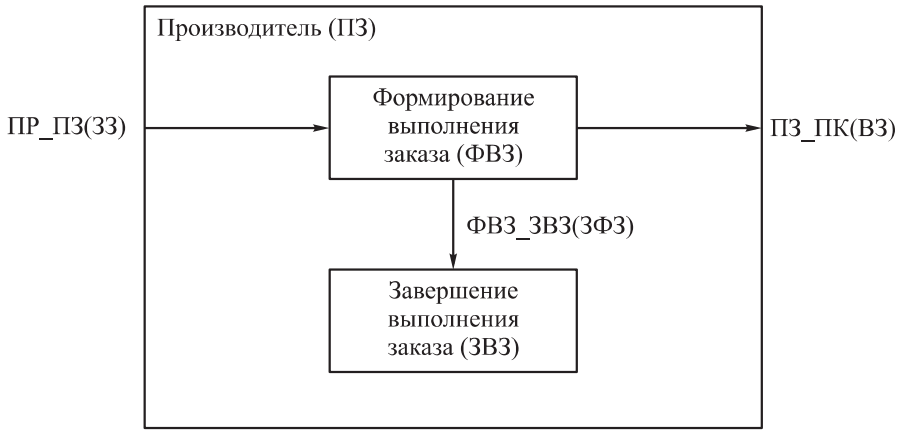

Рис. 5. Агент производитель $A^2_{\text{ПЗ}}$

частично обозначения и понятия, введенные в работах Клини [10] и Роберта Милнера [1]. Каждый агент взаимодействует с другими агентами с помощью действий, называемых восприятиями и реакциями. Восприятие обозначается $?a(?mes)$, где a — имя входного канала $?a$ агента, $(?mes)$ — сообщение, поступающее во входной канал $?a$. Реакция обозначается $!a(!mes)$, где a — имя выходного канала $!a$ агента, $(!mes)$ — сообщение, поступающее в выходной канал $!a$. В общем случае действие (восприятие или реакцию), когда нас не интересует тип канала и сообщение, находящееся в нем, будем обозначать просто символом a с индексом.

Нитью a^* будем называть кортеж (конечный или бесконечный) действий $a^* = a_0a_1a_2 \ldots a_{m-2}a_{m-1}$. Символом $?e$ обозначим пустое восприятие, а символом $!e$ — пустую реакцию. Множество всех нитей, задающих поведение агента, обозначим P^*, алфавит восприятий агента — символами $?A$, а алфавит реакций — как $!A$. Нить конечной длины k, состоящую только из восприятий (нить восприятий), предшествующих какой-либо реакции $!a$, обозначим символами $?a^*(k)$, множество всех таких нитей — как $?A^*$. Нить конечной длины k, состоящую только из реакций (нить реакций), обозначим символами $!a^*(k)$, а множество всех таких нитей — как $!A^*$. Нить конечной длины k, состоящую из последовательностей чередующихся пар $?a!a$, обозначим символами $[?a!a)]^*(k)$, множество всех таких нитей — как $(?A!A)^*$.

Если агент должен выполнять нить конечной длины $a^* = a_0a_1a_2 \ldots a_{m-1}a_m$, то такое поведение агента на языке процессных выра-

жений определяется как процесс $P \triangleq a_0.a_1.a_2. \ldots a_{m-2}.a_m.0.$, где знак $\triangleq$ обозначает равенство по определению; все действия разделяются точками; P — имя процесса; 0 — имя пустого процесса. Наличие пустого процесса после действия a_m означает конец нити a^* (после действия a_m не следует никаких действий). Если нить $a^* \triangleq \triangleq a_1^a a_2^a \ldots a_{m_a}^a a_1^b a_2^b \ldots a_{m_b}^b \ldots a_1^t a_2^t \ldots a_{m_t}^t$ слишком длинная, то вместо процессного выражения $P \triangleq a_1^a.a_2^a. \ldots a_{m_a}^a.a_1^b.a_2^b. \ldots a_{m_b}^b. \ldots a_1^t.a_2^t. \ldots a_{m_t}^t.0.$ может использоваться процессное выражение $P' \triangleq P^a P^b \ldots P^t$, где $P^a \triangleq \triangleq a_1^a.a_2^a. \ldots a_{m_a}^a$, $P^b \triangleq a_1^b a_2^b. \ldots a_{m_b}^b$, $P^t \triangleq a_1^t.a_2^t. \ldots a_{m_t}^t.0.$

Когда процесс имеет общее начало — один и тот же общий подпроцесс $P' \triangleq a_0.a_1.a_2. \ldots a_{m-2}.a_m.$, после выполнения которого могут выполняться различные подпроцессы $P^a \triangleq a_1^a a_2^a \ldots a_{m_a}^a.0.$, $P^b \triangleq a_1^b a_2^b \ldots a_{m_b}^b.0.$, …, $P^t \triangleq a_1^t a_2^t \ldots a_{m_t}^t.0.$, то совокупность процессов $P_a, P_b, ..., P_t$ может быть представлена процессным выражением $P \triangleq P'\left(\left|P^a\right|P^b \ldots\right|P^t\right)$. Вертикальная черта здесь означает, что после процесса P' выполняются те процессы $P^a, P^b, ..., P^t$, у которых выполняются, соответственно, действия a_1^a, $a_1^b, ..., a_1^t$.

Если нить $a^* = a_1 a_2 \ldots a_{m-1} a_m$ зацикливается, то процессное выражение для этого случая выглядит как $P \triangleq \{P'\}$, где $P' \triangleq a_1.a_2. \ldots a_{m-2}.a_m.$ Если зацикливание происходит не с начала, а после некоторой нити a^*, то для такого случая используется процессное выражение $P = a^*.\{P'\}$.

Процессное выражение вида $P \triangleq P^a \left\| P^b \right\| \ldots P^t \left\|\right.$ означает, что процессы $P^a, P^b, ..., P^t$ выполняются параллельно.

Задание процессных выражений, особенно если множество нитей велико, дело непростое. В любом случае сначала надо зафиксировать алфавиты восприятий $?A$ и реакций $!A$. Используя эти алфавиты задавать процессы можно с помощью указанных процессных выражений в зависимости от знаний о бизнес-процессе, имеющихся в нашем распоряжении.

Если использовать для описания поведения агентов любые процессные выражения, рассмотренные ранее, то при такой свободе действий переход от процессных выражений к программной реализации может оказаться непростым. Вообще известно, что чем выразительнее язык, тем

труднее его использовать. Одним из путей обхода подобных трудностей является ограничение выразительности языка до такого уровня, который гарантирует его понимание и упрощает проверку корректности.

Часто удобно ограничиться одним типом процессных выражений вида $P = ?a.!a.P'$, называемых переходами. Процессное выражение $P = ?a.!a.P'$ задает процесс P, выполнение которого состоит в осуществлении восприятия $?a$, затем в выполнении реакции $!a$ и последующем выполнении процесса P'. Для более точного понимания понятия «выполнение процесса» будем полагать, что любой процесс P может находиться в состоянии выполнения и в состоянии покоя. Введение понятия состояний выполнения и покоя позволяет следующим образом уточнить понятие «выполнение процесса», описываемого процессным выражением $P = ?a.!a.P'$: 1) предполагается, что до выполнения перехода, описываемого процессным выражением $P = ?a.!a.P'$, оба процесса P и P' находятся в состоянии покоя; 2) процесс P находится в состоянии покоя до тех пор, пока в результате какого-либо внешнего по отношению к нему действия не перейдет в состояние выполнения; 3) процесс P' остается в состоянии покоя; 4) процесс P находится в состоянии выполнения до тех пор, пока не будут осуществлены сначала восприятие $?a$, а затем реакция $!a$; 5) после этого процесс P переходит в состояние покоя, а процесс P' — в состояние выполнения. Таким образом, процессное выражение $P = ?a.!a.P'$ задает условие или отношение достижимости процессом P процесса P'. Графически переход обычно изображается, как показано на рис. 6.

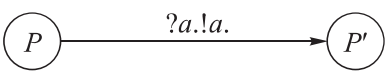

Рис. 6. Графическое изображение перехода

Используя графическое представление перехода, можно любое множество переходов, например, представить в виде процессного графа переходов (рис. 7).

Используя введенные в предыдущих разделах обозначения и понятия, поведение бизнес-процесса реализации заказа может быть описано следующим образом:

$$\text{БПРЗ} \triangleq \text{ПК} \| \text{ПР} \| \text{Б} \| \text{ПЗ},$$

$$\text{ПК} \triangleq \text{ФСЗ} | \text{ПС} | \text{ПЗ} | \text{ЗРПК} |,$$

$$\text{ФСЗ} \triangleq !\text{ПК\_ПР(РЗ)}.!\text{ФСЗ\_ПС(ЗФС)}.\text{ПС} | \; !\text{ПК\_ПР(РЗ)}.!\text{ФСЗ\_ПЗ(ЗПС)}.\text{ПЗ},$$

$$\text{ПС} \triangleq ?\text{ФЗС\_ПС(ЗФС)}.?\text{Б\_ПК(ВС)}.!\text{ПС\_ЗРПК(ЗПС)}.\text{ЗРПК},$$

ПЗ ≜ ?ФСР_ПЗ(ЗФЗ).?ПЗ_ПКВЗ).!ПЗ_ЗРПК(ЗПЗ).ЗРПК,

ЗРПК ≜ ?ПС_ЗРПК(ЗПС).?ПЗ_ЗРПК(ЗПЗ).0

ПР ≜ ФЗС|ФЗЗ|ЗРПР|,

ФЗС ≜ ?ПК_ПР(РЗ).!ПР_Б(ЗС).!ФЗС_ЗРПР(ЗФЗС).ЗРПР,

ФЗЗ ≜ ?ПК_ПР(РЗ).!ПР_ПЗ(ЗЗ).!ФЗЗ_ЗРПР(ЗФЗЗ).ЗРПР,

ЗРПР ≜ ?ФЗС_ЗРПР(ЗФЗС).?ФЗК_ЗРПР(ЗФЗЗ).0

Б ≜ ФС|ЗФС,

ФС ≜ ?ПР_БЗ(ЗС).!Б_ПК(ВС).!ФС_ЗФС(ЗФС).ЗФС,

ЗФ ≜ ?ФС_ЗФС(ЗФС).0

ПЗ ≜ ФВЗ|ЗВЗ,

ФВЗ ≜ ?ПР_ПЗ(ЗЗ).!ПЗ_ПК(ВЗ).!ФВЗ_ЗВЗ(ЗФЗ).ЗВЗ,

ЗВЗ ≜ ?ФВЗ_ЗВЗ(ЗФЗ).0

$$P = P^1 \left| P^2 \right| P^3 \left| P^4, \; P^1 = P_1 | P_2, P^2 = P_3 | P_4, P^3 = P_5 | P_6, P^4 = P_7 | P_8, \right.$$

где

$$P_1 = ?a_1.!a_2.P^2, P_2 = ?a_2.!a_3.P^4,$$
$$P_3 = ?a_1.!a_3.P^3, P_4 = ?a_2.!a_3.P^3$$
$$P_5 = ?a_2.!a_2.P^2, P_6 = ?a_1.!a_3.P^4,$$
$$P_7 = ?a_1.!a_1.P^1, P_8 = ?a_2.!a_1.P^1.$$

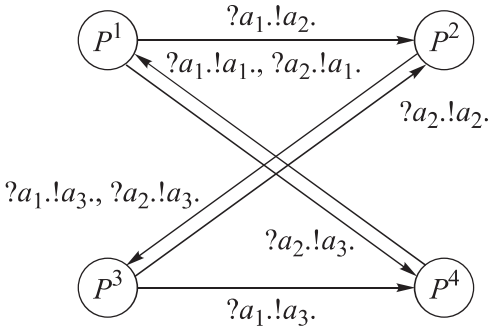

Рис. 7. Процессный граф переходов

Формулировка вопросов для проверки знаний. Формулировку вопросов для проверки знаний будем осуществлять на языках модальных логик вследствие их лаконичности и выразительности применительно к динамическим процессам, каковыми являются бизнес-процессы. Использование той или иной модальной логики зависит от особенностей взаимодействия агентов и знаний, которые необходимо проверить. Выразительные возможности модальных логик возрастают от логики LTL к логике ATL [11].

Принципы формирования вопросов для проверки знаний о поведении агентов будем иллюстрировать на примере поведения бизнес-процесса, приведенного в предыдущем разделе. Наиболее простыми вопросами могут быть, например, вопросы проверки завершения работы. На языке временной модальной логики этими вопросами могут быть следующие [12, 13]:

p(!ПК_ПР(РЗ).ФСЗ) $\supset \Diamond p$(?ПС_ЗРПК(ЗПС).?ПЗ_ЗРПК(ЗПЗ).0)?,

p(?ПК_ПР(РЗ).!ПР_Б(ЗС).ФЗС) $\supset \Diamond p$(?ФЗС_ЗРПР(ЗФЗС).?ФЗК_ЗРПР(ЗФЗЗ).0)?,

p(?ПР_БЗ(ЗС).!Б_ПК(ВС).ФС) $\supset \Diamond p$(?ФС_ЗФС(ЗФС).0)?,

p(?ПР_ПЗ(ЗЗ).!ПЗ_ПК(ВЗ).!ФВЗ_ЗВЗ(ЗФЗ).ЗВЗ) $\supset \Diamond p$(?ФВЗ_ЗВЗ(ЗФЗ).0)?.

Здесь $p(P)$ обозначает предикат, который истинен, когда выполняется процесс P.

Перечисленные однотипные вопросы проверки завершения работы сформулированы на языке линейной временной модальной логики. Ответ на каждый вопрос либо истинен (имеет место соответствующее вопросу завершение работы), либо ложен (соответствующего вопросу завершения работы нет).

Для бизнес-процессов большой размерности, включающих n взаимодействующих агентов, связанных друг с другом m каналами, число пар наборов восприятий и реакций может быть оценено величиной $O(mn2)$. При больших m и n проверка знаний вручную трудно осуществима и требуется автоматизация подобной проверки. Перспективным путем автоматизации является использование логического программирования, например, на основе языка логического программирования Visual Prolog [14]. Принципы создания логических программ, позволяющих автоматизировать подобную проверку, изложен в работе [15].

Заключение. Проверка знаний обучающихся путем анализа построенных ими динамических систем, к числу которых относятся бизнес-процессы, является предметом аналитики в самых различных приложениях и чрезвычайно востребовано на настоящем этапе развития интеллектуальных информационных систем. В настоящей работе изложены принципы проверки знаний о поведении взаимодействующих агентов на основе описания поведения процессными выражениями. Изложение этих принципов осуществлено на примере простого бизнес-процесса. В работе [15] приведены конкретизация подобной проверки для случая доказательства наличия определенных свойств интеллектуальных интерфейсов и реализация этого доказательства на языке логического программирования Visual Prolog. По аналогии этот подход может быть использован и в нашем случае проверки знаний. Поскольку в основе про-

цедур вывода на языке Visual Prolog лежит некоторый вариант унификации, то по отношению к бизнес-процессам унификация может служить средством выявления наличия конкретного поведения. Кроме того, по аналогии с той же работой [15], если известно, какое поведение бизнес-процессов нас наиболее интересует, то вопросы, касающиеся этого поведения, могут формулироваться на языках модальных логик, и затем только они и будут проверяться. Ответы на эти вопросы носят двоичный характер. В дальнейшем предполагается перейти от четкой (двоичной) оценки ответов к нечеткой, используя нечеткую логику. Предполагается программно также воплотить все это для тех проверок знаний на основе бизнес-процессов, которые строит обучающийся.

ЛИТЕРАТУРА

[1] Девятков В.В., Алфимцев А.Н. Необходимые и достаточные формальные свойства мультимодального интерфейса. *Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение*, 2011, спец. вып. «Информационные технологии и компьютерные системы», с. 159–167.

[2] Robin M. Communicating and mobile systems: the π-calculus. Cambridge University Press, 2003.

[3] Medvidovic N., Taylor R.M. A classification and comparison framework for software architecture description languages. *IEEE Trans. Softw. Eng.*, 2000, vol. 26, no. 1, pp. 70–93. DOI: https://doi.org/10.1109/32.825767

[4] Braga C., Sztajnberg A. Towards a rewriting semantics for a software architecture description language. *Electron. Notes Theor. Comput. Sc.,* 2004, vol. 95, pp. 149–168. DOI: https://doi.org/10.1016/j.entcs.2004.04.010

[5] Bruni R., Fiadeiro J.L., Lanese I., et al. New insight into the algebraic properties of architectural connectors. *IFIP TCS*, 2004, vol. 155, pp. 367–380.
DOI: https://doi.org/10.1007/1-4020-8141-3_29

[6] Choutri A., Belala F., Barkaoui K. A tile logic based approach for software architecture description analysis. *JSEA*, 2010, vol. 3, no. 11, pp. 1067–1079.
DOI: http://dx.doi.org/10.4236/jsea.2010.311126

[7] Bouanaka C., Choutri A., Belala F. On generating tile system for a software architecture: case of a collaborative application session. *ICSOFT2007*, 2007, pp. 123–128.

[8] Bog A. A visual environment for the simulation of business processes based on the Pi-calculus. Master Thesis. Potsdam University, 2006.

[9] Девятков В.В. Построение, оптимизация и модификация процессов. *Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение*, 2012, № 4, с. 60–79.

[10] Kleene S.C. Representation of events in nerve nets and finite automata. In: Automata studies. Princeton University Press, 1956, pp. 3–42.

[11] Wooldridge M. An introduction to multiagent systems. John Wiley & Sons, 2009.

[12] Девятков В.В., Румбешт В.В. Нечеткое модальное ситуационное исчисление для анализа сложных объектов. *Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение*, 2001, № 3, с. 3–18.

[13] Леммон Е. Алгебраическая семантика для модальных логик I. II. В: Семантика модальных и интенсиональных логик. М., Прогресс, 1981.

[14] Адаменко А.Н., Кучков А. Логическое программирование и Visual Prolog. СПб., БХВ–Петербург, 2003.

[15] Девятков В.В. Верификация свойств интеллектуальных интерфейсов в логике тайлов. *Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение*, 2016, № 3, с. 65–87. DOI: http://dx.doi.org/10.18698/0236-3933-2016-3-65-87

Девятков Владимир Валентинович — д-р техн. наук, профессор, заведующий кафедрой «Информационные системы и телекоммуникации» МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5, стр. 1).

Кадырбаева Анастасия Рустемовна — ассистентка кафедры «Информационные системы и телекоммуникации» МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5, стр. 1).

Просьба ссылаться на эту статью следующим образом:

Девятков В.В., Кадырбаева А.Р. Верификация знаний, полученных при изучении моделей бизнес-процессов. *Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение*, 2020, № 4, с. 99–113. DOI: https://doi.org/10.18698/0236-3933-2020-4-99-113

VERIFICATION OF KNOWLEDGE OBTAINED IN THE STUDY OF BUSINESS PROCESS MODELS

V.V. Devyatkov deviatkov@bmstu.ru
A.R. Kadyrbaeva naten702@mail.ru

Bauman Moscow State Technical University, Moscow, Russian Federation

Abstract

The paper focuses on the method of formal verification of knowledge obtained in the study of business process models. As a formal language for the presentation and verification of business processes, a certain version of π-calculus was chosen, and linear temporal modal logic was chosen as a language for formulating test assignments, i.e., questions, to be checked. The paper gives the rationale for such choice, and considers the principles of developing the test assignments in the language of modal logic and their usage for knowledge verification. The study proposes to automate knowledge

Keywords

Knowledge verification, test assignment, modal logic, logic programming language Prolog

testing with logic programs obtained as a result of the transition from a description of business processes to a verification program in the logical programming language Visual Prolog. The formal knowledge verification technique is illustrated by examples of knowledge verification. Findings of research show how the transition from the description of business processes to the procedures of formal verification of knowledge is carried out. The prospects for the development of the proposed method are discussed

Received 11.09.2019
Accepted 02.10.2019

This work was supported by the Ministry of Education and Science of the Russian Federation (project no. 2.5048.2017/8.9)

REFERENCES

[1] Devyatkov V.V., Alfimtsev A.N. Necessary and sufficient formal properties of multimodal interface. *Vestn. Mosk. Gos. Tekh. Univ. im. N.E. Baumana, Priborostr.* [Herald of the Bauman Moscow State Tech. Univ., Instrum. Eng.], 2011, spec. iss "Information technology and computer systems", pp. 159–167 (in Russ.).

[2] Robin M. Communicating and mobile systems: the π-calculus. Cambridge University Press, 2003.

[3] Medvidovic N., Taylor R.M. A classification and comparison framework for software architecture description languages. *IEEE Trans. Softw. Eng.*, 2000, vol. 26, no. 1, pp. 70–93. DOI: https://doi.org/10.1109/32.825767

[4] Braga C., Sztajnberg A. Towards a rewriting semantics for a software architecture description language. *Electron. Notes Theor. Comput. Sc.*, 2004, vol. 95, pp. 149–168. DOI: https://doi.org/10.1016/j.entcs.2004.04.010

[5] Bruni R., Fiadeiro J.L., Lanese I., et al. New insight into the algebraic properties of architectural connectors. *IFIP TCS*, 2004, vol. 155, pp. 367–380.
DOI: https://doi.org/10.1007/1-4020-8141-3_29

[6] Choutri A., Belala F., Barkaoui K. A tile logic based approach for software architecture description analysis. *JSEA*, 2010, vol. 3, no. 11, pp. 1067–1079.
DOI: http://dx.doi.org/10.4236/jsea.2010.311126

[7] Bouanaka C., Choutri A., Belala F. On generating tile system for a software architecture: case of a collaborative application session. *ICSOFT2007*, 2007, pp. 123–128.

[8] Bog A. A visual environment for the simulation of business processes based on the Pi-calculus. Master Thesis. Potsdam University, 2006.

[9] Devyatkov V.V. Construction, optimization, and modification of processes. *Vestn. Mosk. Gos. Tekh. Univ. im. N.E. Baumana, Priborostr.* [Herald of the Bauman Moscow State Tech. Univ., Instrum. Eng.], 2012, no. 4, pp. 60–79 (in Russ.).

[10] Kleene S.C. Representation of events in nerve nets and finite automata. In: Automata studies. Princeton University Press, 1956, pp. 3–42.

[11] Wooldridge M. An introduction to multiagent systems. John Wiley & Sons, 2009.

[12] Devyatkov V.V., Rumbesht V.V. Situational modal fuzzy calculus for analysis of complex objects. *Vestn. Mosk. Gos. Tekh. Univ. im. N.E. Baumana, Priborostr.* [Herald of the Bauman Moscow State Tech. Univ., Instrum. Eng.], 2001, no. 3, pp. 3–18 (in Russ.).

[13] Lemmon E. Algebraicheskaya semantika dlya modal'nykh logik I. II [Algebraic semantics for modal logic I. II]. V: Semantika modal'nykh i intensional'nykh logic [In: Semantics of modal and intensional logics]. Moscow, Progress Publ., 1981.

[14] Adamenko A.N., Kuchkov A. Logicheskoe programmirovanie i Visual Prolog [Logic programming and Visual Prolog]. St. Petersburg, BKhV–Peterburg Publ., 2003.

[15] Devyatkov V.V. Verification of intelligent interface properties in the tiles logic. *Vestn. Mosk. Gos. Tekh. Univ. im. N.E. Baumana, Priborostr.* [Herald of the Bauman Moscow State Tech. Univ., Instrum. Eng.], 2016, no. 3, pp. 65–87 (in Russ.).
DOI: http://dx.doi.org/10.18698/0236-3933-2016-3-65-87

Devyatkov V.V. — Dr. Sc. (Eng.), Professor, Head of Department of Information Systems and Telecommunications, Bauman Moscow State Technical University (2-ya Baumanskaya ul. 5, str. 1, Moscow, 105005 Russian Federation).

Kadyrbaeva A.R. — Assistant, Department of Information Systems and Telecommunications, Bauman Moscow State Technical University (2-ya Baumanskaya ul. 5, str. 1, Moscow, 105005 Russian Federation).

Please cite this article in English as:

Devyatkov V.V., Kadyrbaeva A.R. Verification of knowledge obtained in the study of business process models. *Herald of the Bauman Moscow State Technical University, Series Instrument Engineering*, 2020, no. 4, pp. 99–113 (in Russ.).
DOI: https://doi.org/10.18698/0236-3933-2020-4-99-113