МАТЕМАТИЧЕСКИЕ МОДЕЛИ
И ВЫЧИСЛИТЕЛЬНЫЕ СТРУКТУРЫ
(Вычислительные системы)
2004 год Выпуск 173

УДК 519.682.5

О ВЫРАЗИТЕЛЬНОСТИ ЯЗЫКА ОПИСАНИЯ ОПРЕДЕЛЕНИЙ И РЕАЛИЗАЦИЙ ПРОЦЕССОВ[1]

В.А.Васенин[2], А.С.Шундеев [3]

Введение

Исторически, принято считать, что прообразами современных систем автоматизации бизнес–процессов стали появившиеся в 70 х годах прошлого столетия для нужд банковской и страховой сферы системы электронного документооборота [1]. Одно из направлений в развитии электронного документооборота тех лет заключалось в реализации идеи *автоматизированного офиса.* В ее основу было положено понятие потока работ (заданий) между сотрудниками внутри организации. Каждое задание представляет собой совокупность документов и инструкций по их обработке. При этом, система электронного документооборота должна отвечать за правильную и автоматическую навигацию таких заданий между сотрудниками внутри организации.

Другой, также ставшей традиционной, сферой применения системы автоматизации бизнес–процессов является деятель-

[1]Работа выполняется при поддержке Российского фонда фундаментальных исследований, грант 03-07-90117.

[2]e-mail: vasenin@msu.ru

[3]e-mail: shundeev@msu.ru

ность, связанная с *интеграцией приложений* [2]. Существует следующая модель подобной интеграции. Приложения взаимодействуют друг с другом посредством обмена сообщениями. За правильную доставку сообщений между приложениями отвечает посредник (брокер). Если посредник не сохраняет информацию о переданных сообщениях, то он не может использоваться для обработки запросов, направленных от одного приложения к другому. В этом случае он не гарантирует получение ответа на запрос. Посредник в сочетании с системой автоматизации бизнес-процессов может применяться для обработки сложных запросов, в которых одновременно задействованы сразу несколько приложений, обрабатываются промежуточные результаты взаимодействия, на основе которых выбирается дальнейшая стратегия выполнения запроса.

В электронном документообороте и в интеграции приложений используются два полярных типа автоматизированных бизнес-процессов. В первом случае главный акцент ставится на координацию человеческих ресурсов с целью достижения некоторой производственной задачи, а во втором случае — на координацию машинных ресурсов. В настоящее время, на практике, используется широкий диапазон систем автоматизации бизнес-процессов, в которых одновременно и в разной степени задействованы как человеческие, так и машинные ресурсы.

Примером использования таких автоматизированных процессов является новый подход [3] к автоматизации сопровождения *сложных научных экспериментов*, развиваемый в настоящее время в Институте механики МГУ им. М.В. Ломоносова. Под сложным научным экспериментом понимается исследование, проводимое на стыке нескольких научных направлений, в котором задействованы специалисты из разных предметных областей. К числу исследований, которые с полным правом можно рассматривать как сложный научный эксперимент со всеми характерными проблемами, относится задача получения с помощью термомеханического воздействия изделий с требуемыми функциональными свойствами. Данная задача находится на стыке материаловедения, механики, прикладной математики и информатики. Материаловедческий подход к этой задаче заклю-

чается в создании в материале изделия требуемой (регламентируемой) микроструктуры. Проблема может быть сформулирована следующим образом: необходимо разработать термомеханическое воздействие, применение которого к образцу с исходной микроструктурой формирует в материале изделия требуемую микроструктуру. В проведении подобных исследований участвуют специалисты по распознаванию образов, отвечающие за удаление шумов с изображений микроструктуры, материаловеды, технологи, выбирающие в зависимости от типа микроструктуры вид термомеханического воздействия, а также руководитель, планирующий и управляющий ходом проведения эксперимента на всех его этапах. Для успешного выполнения описанной задачи должны быть интегрированы машинные и информационные ресурсы, позволяющие автоматизировать задачи распознавания и классификации микроструктур, а также средства, организующие долгосрочное хранение полученных результатов.

Рассмотрим более подробно понятие бизнес-процесс. Согласно классификатору терминов [4] международной некоммерческой организации Workflow Management Coalition, занимающейся стандартизацией технологий разработки и сопровождения систем автоматизации бизнес-процессов, *бизнес-процесс* --- это одна или более связанных между собой процедур (операций), совместное выполнение которых реализует некоторую производственную задачу. Как правило, подобная производственная задача рассматривается в контексте предприятия или организации. В основе автоматизации бизнес-процесса посредством системы автоматизации бизнес-процессов лежат два понятия: *определение процесса* и *реализация процесса*. Определение процесса представляет собой детально разработанную схему выполнения бизнес-процесса. Система автоматизации бизнес-процессов способна интерпретировать подобную схему с целью создания и управления ходом проведения конкретных реализаций процесса, при необходимости, привлекая человеческие ресурсы, а также, вызывая соответствующие программные приложения и средства. Каждому определению процесса может соответствовать целый набор реализаций, отличающихся друг от друга временем создания и исходными данными.

Одно из главных направлений в развитии технологии систем автоматизации бизнес-процессов связано с проблемами проектирования и создания формальных языков описания и моделирования определений процессов. Каждая система автоматизации бизнес-процессов должна предоставлять подобный язык, например, для ввода в систему новых определений процессов. Кроме того, в последнее время большие усилия уделяются созданию стандартных языков таких, как XPDL [5], предназначенных для обмена определениями процессов между различными системами автоматизации бизнес-процессами. Выделим две главных ситуации, при которых может возникнуть необходимость в подобного рода обмене. Во первых, использование внешних (не поставляемых с рассматриваемой системой автоматизации бизнес процессов) инструментов разработки, моделирования и анализа определений процессов. Во вторых, организация совместного выполнения одной реализации процесса одновременно несколькими различными системами автоматизации бизнес-процессов.

Первоочередная задача в теоретическом исследовании языков описания определений процессов связана с оценкой их выразительных возможностей. Традиционный подход к оценке выразительности средств описания абстрактных вычислительных устройств, выработанный в теории алгоритмов, не может быть применен к рассматриваемому классу языков. Потому, что с традиционной точки зрения языки из рассматриваемого класса равномощны языку машин Тьюринга.

Оригинальный подход к решению данной проблемы был предложен нидерландскими учеными Ван-дер-Альстом и Хофстедом [6,7]. В основу подхода было положено понятие *шаблона* — моделирующей конструкции, поддерживаемой (или нет) средствами тестируемого языка описания определений процессов. Всего было выделено порядка 30 шаблонов, из которых 20, наиболее общих, используются для тестирования. С целью подчеркнуть отличие от традиционного понимания термина *выразительность* Ван-дер-Альстом и Хофстедом было предложено использовать в отношении языков описания определений процессов новый термин *пригодность* (*suitability*). Эффективность указанного подхода была проверена при анализе 15 современных лиди-

рующих систем автоматизации бизнес–процессов и тестировании 7 языков описания определений процессов, претендующих играть роль стандарта. При этом каждый шаблон полностью поддерживается хотя бы одной из систем автоматизации бизнес–процессов, с другой стороны, ни один из языков описания определений процессов не поддерживает полностью все 20 шаблонов. Исследованиям Ван–дер–Альста и Хофстеда посвящен Интернет ресурс [8], который в частности содержит описание используемых в их подходе шаблонов.

Настоящая работа посвящена оценке выразительности (пригодности) разработанного авторами основанного на технологии XML языка описания определений и реализаций процессов XPDIL (XML Process Definition and Instance Language) [9]. Язык XPDIL является основой специализированной системы автоматизации бизнес–процессов, предназначенной для сопровождения проведения сложных научных экспериментов. Главной чертой языка XPDIL является возможность в единообразном виде описывать не только определения процессов, но и внутреннее состояние запущенных реализаций. Данное обстоятельство позволило описать не только синтаксис, но и формализовать семантику языка XPDIL, т.е. формализовать (и доказать) утверждения о (не) поддержке каждого из 20 основных шаблонов языком XPDIL.

На протяжении работы множество натуральных чисел будем обозначать через $\mathbb{N}$. Для введения новых обозначений будем использовать знак $\rightleftharpoons$.

1. Формальная модель

Прежде, чем перейти к исследованию выразительности языка XPDIL, построим формальную модель для тех автоматизированных бизнес–процессов, которые могут быть описаны на данном языке. Начнем с определения процесса.

ОПРЕДЕЛЕНИЕ 1 (определение процесса). *Определением процесса* (*процессом*) назовем набор элементов вида

$$\mathcal{O} = (\mathcal{S}, \mathcal{S}_{a,a}, \mathcal{S}_{a,o}, \mathcal{S}_{o,a}, \mathcal{S}_{o,o}, \mathcal{S}_{\mathcal{S}}, \mathcal{S}_{f}, \mathcal{T}, \mathcal{D}; \sigma, \lambda, \omega), \qquad (1)$$

где

- $\mathcal{S}$ — непустое конечное множество *шагов*;
- $\mathcal{S}_{a,a}, \mathcal{S}_{a,o}, \mathcal{S}_{o,a}, \mathcal{S}_{o,o} \subseteq \mathcal{S}$ — разбиение $\mathcal{S}$ на попарно непересекающиеся множества, удовлетворяющие условию $\mathcal{S} = \mathcal{S}_{a,a} \cup \mathcal{S}_{a,o} \cup \mathcal{S}_{o,a} \cup \mathcal{S}_{o,o}$;
- $\mathcal{S}_s, \mathcal{S}_f \subseteq \mathcal{S}$ — непустое множество *начальных* и непустое множество *заключительных* шагов;
- $\mathcal{T} \subseteq \mathcal{S} \times \mathcal{S}$ — множество *переходов*;
- $\mathcal{D}$ — непустое множество *состояний*;
- $\sigma : \mathcal{T} \longrightarrow \mathbb{N}$ — инъективное отображение, упорядочивающее множество переходов;
- $\lambda : \mathcal{T} \times \mathcal{D} \longrightarrow \{0, 1\}$ — функция, ставящая в соответствие каждому переходу его *условие срабатывания*;
- $\omega : \mathcal{S} \times \mathcal{D} \longrightarrow \mathcal{D}$ — функция *смены состояний*.

В приведенном определении процесс представлен в виде конечного множества шагов $\mathcal{S}$. Каждый шаг представляет собой отдельный «кусок» работы, который должен быть выполнен в рамках процесса. В определении процесса выделены множества начальных $\mathcal{S}_s$ и заключительных $\mathcal{S}_f$ шагов. Выполнение реализации процесса начинается с одновременного выполнения всех начальных шагов и заканчивается завершением выполнения одного из заключительных шагов процесса. Множество шагов $\mathcal{S}$ и множество переходов $\mathcal{T}$ задают графовую структуру процесса. Основным механизмом передачи управления от одних шагов реализации процесса, завершивших свое выполнение, к другим являются сработавшие переходы, а также элементы расщепления и элементы соединения переходов. Функция λ ставит в соответствие каждому переходу его условие срабатывания. Срабатывание перехода заключается в вычислении данного условия.

Выполненный шаг процесса может передать свое управление только через исходящие переходы. За выбор тех исходящих переходов, через которые будет передано управление, отвечает

элемент расщепления, приписанный шагу процесса. Аналогично, шаг процесса может начать свое выполнение только после передачи ему управления через входящие переходы. За выбор входящих переходов отвечает элемент соединения, приписанный шагу процесса. В предлагаемой формальной модели элементы расщепления и элементы соединения могут быть двух типов — AND и XOR. Отличие друг от друга этих типов будет показано ниже. Множества $\mathcal{S}_{a,a}$, $\mathcal{S}_{a,o}$, $\mathcal{S}_{o,a}$ и $\mathcal{S}_{o,o}$ задают типы элементов соединения и расщепления шагов процесса. Первый индекс определяет тип элемента соединения, а второй индекс определяет тип элемента расщепления для всех шагов процесса в соответствующем множестве. Символ a соответствует типу AND, а символ o — типу XOR.

Отметим ряд допущений, реализованных в предложенной модели определения процесса. Первое допущение связано с отказом от привязки к конкретному способу обработки данных (ручная обработка, автоматическая обработка или вызов подпроцесса) при выполнении шагов процесса. Для этого была введена функция смены состояний ω, которая меняет внутреннее состояние процесса $\delta \in \mathcal{D}$ на $\omega(s, \delta)$ при выполнении его шага $s \in \mathcal{S}$. Второе допущение относится к представлению внутреннего состояния процесса. Вместо использования конечного набора данных и их строковых значений было введено непустое множество состояний $\mathcal{D}$. Каждый элемент $\delta \in \mathcal{D}$ целиком задает внутреннее состояние процесса.

Язык XPDIL представляет собой средство документирования не только определений процессов, но также и промежуточных этапов выполнения реализаций процессов. Данное обстоятельство должно быть отражено в формальной модели.

ОПРЕДЕЛЕНИЕ 2 (этап выполнения реализации процесса). *Этапом выполнения реализации процесса* (*этапом*) назовем набор элементов вида

$$\mathcal{P} = (\mathcal{O}; \mathcal{S}_r, \mathcal{S}_c, \mathcal{T}_c, \mathcal{T}_a; \delta), \tag{2}$$

где

- $\mathcal{O}$ — определение процесса вида (1);

- $\mathcal{S}_r$, $\mathcal{S}_c \subseteq \mathcal{S}$ — непересекающиеся множества *выполняющихся* и *выполненных* шагов;
- $\mathcal{T}_c$, $\mathcal{T}_a \subseteq \mathcal{T}$ — непересекающиеся множества *сработавших* и *неуспешно сработавших* переходов;
- $\delta \in \mathcal{D}$ — *внутреннее состояние*.

Множество всех этапов, соответствующих определению процесса $\mathcal{O}$, будем обозначать через $\mathbb{P}(\mathcal{O})$.

ЗАМЕЧАНИЕ. В дальнейшем, мы будем придерживаться следующих соглашений относительно обозначений для определений процессов и этапов выполнения реализаций процессов. Если не оговорено противное, то для составных компонент процесса, обозначенного символом $\mathcal{O}$, и этапа, обозначенного символом $\mathcal{P}$, будем использовать соответственно обозначения (1) и (2). Проиндексированные натуральными числами этапы, соответствующие определению процесса $\mathcal{O}$, будем обозначать через $\mathcal{P}_i = (\mathcal{O}; \mathcal{S}_r^i, \mathcal{S}_c^i; \mathcal{T}_c^i, \mathcal{T}_a^i; \delta_i)$ (i — натуральный индекс).

В рамках этапа выполнения реализации процесса $\mathcal{P}$ выделенные множества $\mathcal{S}_r$, $\mathcal{S}_c$ и $\mathcal{S} \setminus \{\mathcal{S}_r \cup \mathcal{S}_c\}$ задают три возможных состояния, в которых может находится шаг процесса $s \in \mathcal{S}$. Данные состояния будем обозначать соответственно через (r), (c) и (n). Относительно шага, находящегося в состоянии (n), будем говорить, что он не начал свое выполнение. Аналогично, выделенные множества $\mathcal{T}_c$, $\mathcal{T}_a$ и $\mathcal{T} \setminus \{\mathcal{T}_c \cup \mathcal{T}_a\}$ задают три возможных состояния, в которых может находится переход процесса $t \in \mathcal{T}$. Данные состояния будем обозначать соответственно через (c), (a) и (n). Относительно перехода, находящегося в состоянии (n), будем говорить, что он не подвергался процедуре срабатывания.

Особое положение при выполнении реализации процесса занимают этапы, которые интерпретируются как начальные и заключительные.

ОПРЕДЕЛЕНИЕ 3 (начальный этап). Этап $\mathcal{P} \in \mathbb{P}(\mathcal{O})$ назовем *начальным*, если $\mathcal{S}_r \cup \mathcal{S}_c = \varnothing$ и $\mathcal{T}_c \cup \mathcal{T}_a = \varnothing$. Множество всех начальных этапов, соответствующих определению процесса $\mathcal{O}$, будем обозначать через $\mathbb{S}(\mathcal{O})$.

Из приведенного определения вытекает, что для начального этапа необходимо, чтобы ни один шаг процесса не был выполнен

и не находится в состоянии выполнения, а также, чтобы ни к одному переходу процесса не применялась процедура срабатывания. От заключительного этапа достаточно потребовать завершение выполнения заключительного шага.

ОПРЕДЕЛЕНИЕ 4 (заключительный этап). Этап $\mathcal{P} \in \mathbb{P}(\mathcal{O})$ назовем *заключительным*, если $\mathcal{S}_c \cap \mathcal{S}_f \neq \varnothing$. Множество всех заключительных этапов, соответствующих определению процесса $\mathcal{O}$, будем обозначать через $\mathbb{F}(\mathcal{O})$.

Введенные понятия определения процесса, этапа выполнения реализации процесса, а также понятия начального и заключительного этапов образуют статическую составляющую формальной модели языка XPDIL. Динамическая составляющая должна включать в себя набор отношений, описывающих для каждого конкретного шага процесса возможность его перехода и сам переход из одного состояния в другое во время выполнении реализации процесса.

Рассмотрим первую пару отношений, описывающих возможность выполнения и начало выполнения шага процесса.

ОПРЕДЕЛЕНИЕ 5 (отношение $\in_n$). Предположим, что

- $\mathcal{P} \in \mathbb{P}(\mathcal{O}) \setminus \mathbb{F}(\mathcal{O})$;
- $s \in \mathcal{S} \setminus \{\mathcal{S}_r \cup \mathcal{S}_c\}$;
- $i(s) \rightleftharpoons \{\, (s_1, s_2) \in \mathcal{T} \mid s_1 \neq s_2,\, s_2 = s \,\}$.

Будем говорить, что шаг s *обладает возможностью запуска* на этапе $\mathcal{P}$ и записывать $s \in_n \mathcal{P}$, если выполняется хотя бы одно из ниже следующих условий:

- $s \in \mathcal{S}_s$;
- $s \in \mathcal{S}_{a,\,a} \cup \mathcal{S}_{a,\,o}$ и $i(s) \neq \varnothing$, $i(s) \subseteq \mathcal{T}_c$;
- $s \in \mathcal{S}_{o,\,a} \cup \mathcal{S}_{o,\,o}$ и $i(s) \cap \mathcal{T}_c \neq \varnothing$.

Как видно из приведенного определения, независимо от типа элемента соединения, возможностью запуска обладают только начальные шаги процесса. Для не начального шага процесса с элементом соединения типа AND возможность запуска равносильна срабатыванию всех отличных от петель входящих в него переходов. В то же время, для не начального шага процесса с

элементом соединения типа XOR достаточно срабатывания хотя бы одного перехода.

ОПРЕДЕЛЕНИЕ 6 (отношение $[s\rangle_n$). Каждому шагу процесса $s \in \mathcal{S}$ поставим в соответствие заданное на множестве $\mathbb{P}(\mathcal{O})$ бинарное отношение $[s\rangle_n$, которое назовем *отношением запуска*. По определению положим $\mathcal{P}\,[s\rangle_n\,\mathcal{P}'$, если

- $s \in_{\mathrm{n}} \mathcal{P}$;
- $\mathcal{P}' = (\mathcal{O};\ \mathcal{S}_r \cup \{s\},\ \mathcal{S}_c,\ \mathcal{T}_c,\ \mathcal{T}_a;\ \delta)$.

Отношение запуска $[s\rangle_n$ переводит шаг s из состояния (n) в состояние (r). В то же время другие отличные от s шаги и переходы процесса не меняют своего состояния.

Следующая пара отношений описывает возможность завершения выполнения и завершение выполнения шага процесса.

ОПРЕДЕЛЕНИЕ 7 (отношение $\in_r$). Предположим, что

- $\mathcal{P} \in \mathbb{P}(\mathcal{O}) \setminus \mathbb{F}(\mathcal{O})$;
- $s \in \mathcal{S}_r$.

Будем говорить, что шаг s *обладает возможностью завершения выполнения* на этапе $\mathcal{P}$ и записывать $s \in_r \mathcal{P}$.

ОПРЕДЕЛЕНИЕ 8 (отношение $[s\rangle_r$). Поставим в соответствие каждому шагу процесса $s \in \mathcal{S}$ заданное на множестве $\mathbb{P}(\mathcal{O})$ бинарное отношение $[s\rangle_r$, которое назовем *отношением завершения выполнения*. По определению положим $\mathcal{P}\,[s\rangle_r\,\mathcal{P}'$, если

- $s \in_{\mathrm{r}} \mathcal{P}$;
- $\mathcal{P}' = (\mathcal{O};\ \mathcal{S}_r \setminus \{s\},\ \mathcal{S}_c \cup \xi(s,\ \mathcal{P}),\ \mathcal{T}_c,\ \mathcal{T}_a;\ \omega(s, \delta))$,

где

$$\xi(s,\ \mathcal{P}) \rightleftharpoons \begin{cases} \varnothing, & \text{если} \quad (s, s) \in \mathcal{T} \wedge \lambda(s, s, \omega(s, \delta)); \\ \{s\}, & \text{в противном случае.} \end{cases}$$

Функция ξ позволяет различать два способа выполнения шага процесса. Если шаг связан петлей (переход, который одновременно является и исходящим, и входящим для шага), то он интерпретируется как тело цикла. Вычисление условия, приписанного петле, определяет возможность повторного выполнения шага процесса в качестве очередной итерации цикла.

Отношение завершения выполнения $[s\rangle_{\mathrm{r}}$ переводит шаг s из состояния (r) либо в состояние (n), либо в состояние (c). При этом отличные от s шаги и переходы процесса не меняют своего состояния.

Последняя пара отношений описывает возможность передачи управления и саму передачу управления после выполнения шага процесса.

ОПРЕДЕЛЕНИЕ 9 (отношение $\in_c$). Предположим, что

- $\mathcal{P} \in \mathbb{P}(\mathcal{O}) \setminus \mathbb{F}(\mathcal{O})$;
- $s \in \mathcal{S}_c$;
- $o(s) \rightleftharpoons \{\, (s_1, s_2) \in \mathcal{T} \mid s_1 \neq s_2,\ s_1 = s \,\}$;
- $o(s) \cap \{\mathcal{T}_c \cup \mathcal{T}_a\} = \varnothing$.

Будем говорить, что шаг s *обладает возможностью передачи управления* на этапе $\mathcal{P}$ и записывать $s \in_c \mathcal{P}$.

Прежде, чем перейти к определению отношения передачи управления, введем для каждого шага процесса процедуру срабатывания исходящих из него переходов. Результат данной процедуры зависит от типа элемента расщепления шага процесса.

ОПРЕДЕЛЕНИЕ 10 (процедура срабатывания переходов). Для произвольных $s \in \mathcal{S}$ и $\delta \in \mathcal{D}$ введем следующие множества переходов

- $p(s, \delta, t) \rightleftharpoons \{\, t' \in o(s) \mid \sigma(t') < \sigma(t) \,\}$ для всех $t \in o(s)$;
- $f_1(s, \delta) \rightleftharpoons \{\, t \in o(s) \mid \lambda(t, \delta) \,\}$;
- $f_2(s, \delta) \rightleftharpoons \{\, t \in o(s) \mid \lambda(t, \delta) \wedge \forall t' \in p(s, \delta, t) : \neg\lambda(t', \delta) \,\}$.

Вычисление множества

$$\mathcal{T}_w(s, \delta) \rightleftharpoons \begin{cases} f_1(s, \delta), & \text{если} \quad s \in \mathcal{S}_{a,\,a} \cup \mathcal{S}_{o,\,a}; \\ f_2(s, \delta), & \text{если} \quad s \in \mathcal{S}_{a,\,o} \cup \mathcal{S}_{o,\,o} \end{cases}$$

назовем *процедурой срабатывания переходов*, исходящих из шага s.

ОПРЕДЕЛЕНИЕ 11 (отношение $[s\rangle_c$). Поставим в соответствие каждому шагу процесса $s \in \mathcal{S}$ заданное на множестве $\mathbb{P}(\mathcal{O})$

бинарное отношение $[s\rangle_c$, которое назовем *отношением передачи управления*. По определению положим $\mathcal{P}\,[s\rangle_c\,\mathcal{P}'$, если

- $s \in_c \mathcal{P}$;
- $\mathcal{P}' = (\mathcal{O};\ \mathcal{S}_r,\ \mathcal{S}_c,\ \mathcal{T}_c \cup \mathcal{T}_w(s,\delta),\ \mathcal{T}_a \cup \{o(s) \setminus \mathcal{T}_w(s,\delta)\};\ \delta)$.

Отношение передачи управления $[s\rangle_c$ не меняет состояние шагов процесса. Тем не менее происходит изменение состояний переходов, исходящих из шага s. Состояние (n) исходящего перехода изменяется либо на состояние (c), либо на состояние (a). Состояние других переходов процесса не изменяется.

Следующее предложение подчеркивает отличия введенных отношений.

ПРЕДЛОЖЕНИЕ 1. *Для каждого шага процесса $s \in \mathcal{S}$ отношения $[s\rangle_n$, $[s\rangle_r$, $[s\rangle_c$ попарно не пересекаются.*

ДОКАЗАТЕЛЬСТВО. Пусть $\mathcal{P}_1, \mathcal{P}_2 \in \mathbb{P}(\mathcal{O})$ — два этапа, соответствующие определению процесса $\mathcal{O}$. Рассмотрим для них три попарно несовместных условия

1) $\mathcal{S}_r^2 = \mathcal{S}_r^1 \cup \{s\} \wedge s \notin \mathcal{S}_r^1$;
2) $\mathcal{S}_r^2 = \mathcal{S}_r^1 \setminus \{s\}$;
3) $\mathcal{S}_r^2 = \mathcal{S}_r^1$.

Учитывая тот факт, что первое условие является необходимым для выполнения отношения $\mathcal{P}_1\,[s\rangle_n\,\mathcal{P}_2$, второе условие — для отношения $\mathcal{P}_1\,[s\rangle_r\,\mathcal{P}_2$, а третье условие — для отношения $\mathcal{P}_1\,[s\rangle_c\,\mathcal{P}_2$, получим доказательство предложения. □

ОПРЕДЕЛЕНИЕ 12 (отношения $[s\rangle$ и $[\,\rangle$). *Отношением срабатывания* шага процесса $s \in \mathcal{S}$ назовем объединение $[s\rangle \rightleftharpoons [s\rangle_n \cup [s\rangle_r \cup [s\rangle_c$.

Объединение $[\,\rangle \rightleftharpoons \bigcup_{s \in \mathcal{S}} [s\rangle$ назовем *отношением смены состояний.*

В случае, когда выполняется отношение $\mathcal{P}_1\,[s\rangle\,\mathcal{P}_2$, будем говорить, что в результате срабатывания отношения $[s\rangle$ произошел переход от этапа $\mathcal{P}_1$ выполнения реализации процесса к этапу $\mathcal{P}_2$. Аналогично будем говорить и о срабатывании отношений $[s\rangle_n$, $[s\rangle_r$, $[s\rangle_c$ и $[\,\rangle$.

ПРЕДЛОЖЕНИЕ 2. *Для различных шагов процесса $s_1, s_2 \in \mathcal{S}$ отношения $[s_1\rangle$ и $[s_2\rangle$ не пересекаются.*

ДОКАЗАТЕЛЬСТВО. Действительно, если выполняется отношение $\mathcal{P}_1 [s_1\rangle \mathcal{P}_2$, то это означает, что при переходе от этапа $\mathcal{P}_1$ к этапу $\mathcal{P}_2$ происходит либо изменение состояние шага s_1, либо изменение состояния исходящих из шага s_1 переходов. При этом состояние шага s_2 и исходящих из него переходов не меняется. Следовательно, $[s_1\rangle \cap [s_2\rangle = \varnothing$. □

Переход от использования отношений $[s\rangle_n$, $[s\rangle_r$, $[s\rangle_c$ к отношению смены состояний $[\,\rangle$ не несет в себе потери информации о произошедшей смене этапов выполнения реализации процессов. Справедливо следующее предложение.

ПРЕДЛОЖЕНИЕ 3. *Предположим, что этапы* $\mathcal{P}_1, \mathcal{P}_2 \in \mathbb{P}(\mathcal{O})$ *связаны отношением* $\mathcal{P}_1 [\,\rangle \mathcal{P}_2$. *Тогда существует ровно один шаг* $s \in \mathcal{S}$ *и ровно один символ* $x \in \{n, r, c\}$, *для которых* $\mathcal{P}_1 [s\rangle_x \mathcal{P}_2$.

ДОКАЗАТЕЛЬСТВО непосредственно следует из предложений 1 и 2. □

Введенное отношение смены состояний $[\,\rangle$ позволяет ввести понятия регулярного этапа и реализации процесса.

ОПРЕДЕЛЕНИЕ 13 (регулярный этап). Рассмотрим последовательность этапов выполнения реализации процесса вида

$$\mathcal{P}_i \in \mathbb{P}(\mathcal{O}), \quad i = 1, \ldots, l, \tag{3}$$

в которой

- $\mathcal{P}_i [\,\rangle_c \mathcal{P}_{i+1}, \quad 1 \leqslant i \leqslant l-1$;
- $\mathcal{P}_1 \in \mathbb{S}(\mathcal{O})$.

Каждый элемент последовательности (3) будем называть *регулярным* этапом.

На основе понятия регулярного этапа формализуем понятие реализации процесса.

ОПРЕДЕЛЕНИЕ 14 (реализация процесса). Пусть $\mathcal{R}$ — последовательность регулярных этапов вида (3), для которой справедливы предположения из предыдущего определения и $\mathcal{P}_l \in \mathbb{F}(\mathcal{O})$. Последовательность $\mathcal{R}$ назовем *реализацией процесса* (*реализацией*) типа $(a, b; l)$, где

- $a = \delta_1$ — *исходные* (*начальные*) *данные*;

- $b = \delta_l$ — *результат*;
- $l \geqslant 2$ — *размер реализации.*

Прежде, чем перейти к рассмотрению примеров, остановимся более подробно на ряде свойств реализаций процессов. Для этого введем необходимую терминологию.

Из определения реализации процесса $\mathcal{R}$ следует, что каждая пара соседних промежуточных состояний $\mathcal{P}_i$ и $\mathcal{P}_{i+1}$, $1 \leqslant i \leqslant l-1$, связана единственным отношением вида $[s\rangle_x$, где $s \in \mathcal{S}$ и $x \in \{n, r, c\}$. Будем говорить, что реализация $\mathcal{R}$ содержит срабатывание отношения $[s\rangle_x$, а число i интерпретировать как порядок (момент) данного срабатывания. Отметим, что в некоторых случаях, срабатывание отношения $[s\rangle_x$ может произойти более одного раза.

Первое свойство связано с типом отношений, которые могут сработать в начале и в конце реализации. Самым первым должно быть срабатывание отношения запуска, а самым последним — срабатывание отношения завершения выполнения.

ПРЕДЛОЖЕНИЕ 4 (первое и последнее срабатывание). *В реализации $\mathcal{R}$ вида* (3) *выполняются условия* $\mathcal{P}_1\,[s\rangle_n\,\mathcal{P}_2$ *и* $\mathcal{P}_{l-1}\,[s'\rangle_r\,\mathcal{P}_l$ *для некоторых шагов* $s, s' \in \mathcal{S}$.

ДОКАЗАТЕЛЬСТВО. В качестве s и s' выберем те шаги, при которых $\mathcal{P}_1\,[s\rangle\,\mathcal{P}_2$ и $\mathcal{P}_{l-1}\,[s'\rangle\,\mathcal{P}_l$. Напомним, что данные шаги определяются однозначно.

Первый этап $\mathcal{P}_1$ реализации $\mathcal{R}$ является начальным. Следовательно, множества выполняющихся и выполненных шагов у этапа $\mathcal{P}_1$ являются пустыми, $\mathcal{S}_r^1 \cup \mathcal{S}_c^1 = \varnothing$. Поэтому на этапе $\mathcal{P}_1$ отношения $[s\rangle_r$ и $[s\rangle_c$ сработать не могут. Действительно, необходимым условием для срабатывания отношения $[s\rangle_r$ является выполнение включения $s \in \mathcal{S}_r^1$, а для срабатывания $[s\rangle_c$ — выполнение включения $s \in \mathcal{S}_c^1$. Учитывая, что $[s\rangle$ является объединением $[s\rangle_n$, $[s\rangle_r$ и $[s\rangle_c$, получим $\mathcal{P}_1\,[s\rangle_n\,\mathcal{P}_2$.

Отношение $[s'\rangle$ может сработать только при не заключительном состоянии $\mathcal{P}_{l-1}$, у которого множество выполненных шагов не содержит заключительного шага, $\mathcal{S}_c^{l-1} \cap \mathcal{S}_f = \varnothing$. В то же время, промежуточное состояние $\mathcal{P}_l$ является заключительным. Поэтому $\mathcal{S}_c^l \cap \mathcal{S}_f \neq \varnothing$, и, следовательно, $\mathcal{S}_c^{l-1} \neq \mathcal{S}_c^l$. Множе-

ство выполненных шагов может измениться только в результате срабатывания отношения завершения выполнения. Следовательно, $\mathcal{P}_{l-1}\,[s']\rangle_r\,\mathcal{P}_l$. □

Из доказательства предложения 4 вытекает следующее утверждение.

СЛЕДСТВИЕ 1. *В реализации процесса последним срабатывает заключительный шаг.*

Перейдем к исследованию порядка срабатывания отношений разных типов в реализации процесса.

ПРЕДЛОЖЕНИЕ 5 (порядок срабатывания отношений). *Пусть $\mathcal{R}$ — реализация вида* (3), *содержащая для некоторого шага $s \in \mathcal{S}$ срабатывание отношения $[s\rangle_c$. Тогда срабатыванию отношения $[s\rangle_c$ в реализации $\mathcal{R}$ должно предшествовать сначала срабатывание отношения $[s\rangle_n$, а затем срабатывание отношения $[s\rangle_r$.*

ДОКАЗАТЕЛЬСТВО. По условию, реализация $\mathcal{R}$ содержит пару этапов вида $\mathcal{P}_h\,[s\rangle_c\,\mathcal{P}_{h+1}$, $1 \leqslant h < l$. Условие $s \in_c \mathcal{P}_h$ является необходимым для срабатывания отношения $[s\rangle_c$. Поэтому, справедливо включение $s \in \mathcal{S}_c^h$. Первый этап $\mathcal{P}_1$ является начальным. По определению начального этапа $s \notin \mathcal{S}_r^1 \cup \mathcal{S}_c^1$. Но тогда шаг s может стать выполненным только в результате срабатывания отношения $[s\rangle_r$. Таким образом, $\mathcal{P}_q\,[s\rangle_r\,\mathcal{P}_{q+1}$ для некоторого q, $1 \leqslant q < h < l$. Следовательно, в реализации $\mathcal{R}$ срабатывание отношения $[s\rangle_r$ предшествует срабатыванию $[s\rangle_c$.

Повторяя проведенные выше рассуждения, получим, что необходимым условием срабатывания отношения $[s\rangle_r$ является включение $s \in \mathcal{S}_r^q$, которому должно предшествовать срабатывание отношения $[s\rangle_n$. □

Сформулируем следствие, вытекающее из доказательства предложения 5.

СЛЕДСТВИЕ 2. *В реализации $\mathcal{R}$ для каждого шага $s \in \mathcal{S}$ отношение $[s\rangle_c$ не может сработать более одного раза.*

ПРИМЕР 1. В качестве примера рассмотрим определение процесса

$$\mathcal{O}_\Pi = (\mathcal{S},\, \mathcal{S}_{a,\,a},\, \mathcal{S}_{a,\,o},\, \mathcal{S}_{o,\,a},\, \mathcal{S}_{o,\,o},\, \mathcal{S}_S,\, \mathcal{S}_f,\, \mathcal{T},\, \mathcal{D};\, \sigma,\, \lambda,\, \omega),$$

состоящее из трех шагов $\mathcal{S} = \{s_1, s_2, s_3\}$, соединенные при помощи двух переходов $\mathcal{T} = \{(s_1, s_2), (s_1, s_3)\}$. Процесс $\mathcal{O}_\Pi$ имеет

единственный начальный шаг $S_S = \{s_1\}$ и два заключительных шага $S_f = \{s_2, s_3\}$. Все шаги имеют элементы соединения и расщепления типа AND, $S_{a,\,a} = S$.Порядок переходов фиксирован следующим образом $\sigma(s_1, s_2) = 1$ и $\sigma(s_1, s_3) = 2$. Переходам приписаны тождественно истинные условия $\lambda(s_1, s_2) = \lambda(s_1, s_3) = 1$. В качестве множества состояний процесса используется множество натуральных чисел $\mathcal{D} = \mathbb{N}$. Функция смены состояний определена по правилу

$$\omega(s, \delta) = \begin{cases} \delta, & \text{если} \quad s = s_1; \\ 2, & \text{если} \quad s = s_2; \\ 3, & \text{если} \quad s = s_3. \end{cases}$$

При любом начальном этапе $\mathcal{P}_1$ (произвольных исходных данных $\delta \in \mathbb{N}$) начало любой реализации рассматриваемого процесса $\mathcal{O}_\text{п}$ будет иметь вид

$$\begin{array}{lll} \mathcal{P}_1 = (\mathcal{O}_\text{п};\, \varnothing,\, \varnothing,\, \varnothing,\, \varnothing;\, \delta), & & \\ \mathcal{P}_2 = (\mathcal{O}_\text{п};\, \{s_1\},\, \varnothing,\, \varnothing,\, \varnothing;\, \delta) & & (\,\mathcal{P}_1\ [s_1\rangle_n\ \mathcal{P}_2\,), \\ \mathcal{P}_3 = (\mathcal{O}_\text{п};\, \varnothing,\, \{s_1\},\, \varnothing,\, \varnothing;\, \delta) & & (\,\mathcal{P}_2\ [s_1\rangle_r\ \mathcal{P}_3\,), \\ \mathcal{P}_4 = (\mathcal{O}_\text{п};\, \varnothing,\, \{s_1\},\, \{(s_1, s_2), (s_1, s_3)\},\, \varnothing;\, \delta) & & (\,\mathcal{P}_3\ [s_1\rangle_c\ \mathcal{P}_4\,). \end{array}$$

Действительно, процесс $\mathcal{O}_\text{п}$ имеет только один начальный шаг s_1. Следовательно, на начальном этапе $\mathcal{P}_1$ может сработать только отношение $[s_1\rangle_n$, порождая этап $\mathcal{P}_2$. При этом шаг s_1 начинает свое выполнение. Этап $\mathcal{P}_2$ может измениться только за счет завершения выполнения шага s_1, что соответствует срабатыванию отношения $[s_1\rangle_r$ и переходу к этапу $\mathcal{P}_3$. Далее, этап $\mathcal{P}_3$ позволяет применить к шагу s_1 процедуру срабатывания исходящих переходов за счет срабатывания отношения $[s_1\rangle_c$. В результате осуществляется переход к промежуточному этапу $\mathcal{P}_4$, который подразумевает две альтернативы

$$\begin{array}{ll} \mathcal{P}_5 = (\mathcal{O}_\text{п};\, \{s_2\},\, \{s_1\},\, \{(s_1, s_2), (s_1, s_3)\},\, \varnothing;\, \delta) & (\,\mathcal{P}_4\ [s_2\rangle_n\ \mathcal{P}_5\,), \\ \mathcal{P}_6 = (\mathcal{O}_\text{п};\, \{s_3\},\, \{s_1\},\, \{(s_1, s_2), (s_1, s_3)\},\, \varnothing;\, \delta) & (\,\mathcal{P}_4\ [s_3\rangle_n\ \mathcal{P}_6\,). \end{array}$$

Одновременно возможно срабатывание отношений $[s_2\rangle_n$ и $[s_3\rangle_n$. В первом случае, происходит запуск на выполнение шага s_2 и

переход к этапу $\mathcal{P}_5$. Во втором случае, происходит запуск на выполнение шага s_3 и соответственно переход к $\mathcal{P}_6$. Далее, возможен переход к заключительным этапам

$$\mathcal{P}_7 = (\mathcal{O}_\text{п};\ \varnothing,\ \{s_1, s_2\},\ \{(s_1, s_2), (s_1, s_3)\},\ \varnothing;\ 2) \qquad (\mathcal{P}_5\ [s_2\rangle_r\ \mathcal{P}_7),$$
$$\mathcal{P}_8 = (\mathcal{O}_\text{п};\ \varnothing,\ \{s_1, s_3\},\ \{(s_1, s_2), (s_1, s_3)\},\ \varnothing;\ 3) \qquad (\mathcal{P}_6\ [s_3\rangle_r\ \mathcal{P}_8).$$

В заключение отметим, что две получившиеся реализации, которые будем обозначать через $\mathcal{R}_1$ и $\mathcal{R}_2$, имеют одинаковые исходные данные и разные результаты соответственно 2 и 3.

2. Исследование выразительности

Перейдем к исследованию выразительности (пригодности) языка XPDIL на основе подхода, разработанного Ван-дер-Альстом и Хофстедом. Данный подход предполагает исследование возможности реализовать конструкцию каждого из двадцати стандартных шаблонов средствами языка XPDIL. Результат подобного исследования заключается в возможности для каждого из стандартных шаблонов установить одну из трех возможных альтернатив: шаблон поддерживается (имеет прямую реализацию), шаблон не поддерживается, шаблон имеет частичную реализацию в языке XPDIL. На сегодняшний день исследована возможность реализации каждого из 20 стандартных шаблонов в языке XPDIL. В настоящей работе мы рассмотрим наиболее важные, с точки зрения языка XPDIL, шаблоны.

В предстоящем исследовании существенную роль будет играть введенная ранее формальная модель языка XPDIL. В рамках данной модели будут формулироваться и доказываться утверждения о поддержке тех или иных шаблонов. Для большинства не поддерживаемых языком XPDIL шаблонов соответствующие утверждения не могут быть формализованы в рамках модели языка XPDIL. Действительно, с этой целью необходимо использовать более «широкую» модель, существенно расширяя формальную модель языка XPDIL. Чтобы обойти данное ограничение, для каждого неподдерживаемого шаблона будет формулироваться возможное необходимое условие его поддержки для

дальнейшего опровержения. Как правило, при помощи построения контрпримера. Для шаблонов, имеющих частичную реализацию, должна быть показана невозможность их прямой реализации. Кроме того, для них необходимо указать обоснованную заменяющую конструкцию в терминах формальной модели языка XPDIL.

Начнем с шаблона WP1, реализующего конструкцию последовательности. Здесь и далее, для тестируемых шаблонов мы будем использовать обозначения из работы [6] вида WPn, где WP является сокращением от Workflow Pattern, а n — порядковый номер шаблона. Данная конструкция заключается в последовательном выполнении набора шагов процесса. При этом порядок их выполнения фиксирован, и в каждый момент времени может выполняться только один шаг.

ОПРЕДЕЛЕНИЕ 15 (тестовый процесс для шаблона WP1). Определение процесса $\mathcal{O}$ с упорядоченным множеством шагов $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$, $n \in \mathbb{N}$, назовем *тестовым процессом для шаблона* WP1 *размера* n, если для любой реализации данного процесса $\mathcal{R}$ вида (3) выполняются условия

- $\operatorname{card} \mathcal{S}_r^i \leqslant 1, \quad i = 1, \ldots, l;$
- $s_j \in \mathcal{S}_c^i \Longrightarrow s_{j-1} \in \mathcal{S}_c^i, \quad j = 2, \ldots, n.$

Шаблон WP1 напрямую поддерживается языком XPDIL.

ПРЕДЛОЖЕНИЕ 6 (поддержка шаблона WP1) *Для каждого натурального* $n \geqslant 2$ *существует тестовый для шаблона* WP1 *процесс* $\mathcal{O}$ *размера* n.

ДОКАЗАТЕЛЬСТВО. Искомый тестовый процесс $\mathcal{O}$ определим по правилам

- $\mathcal{S}_{a,a} = \mathcal{S}$;
- $\mathcal{S}_S = \{s_1\}$, $\mathcal{S}_f = \{s_n\}$;
- $\mathcal{T} = \{(s_{j-1}, s_j) \mid 2 \leqslant j \leqslant n\}$;
- λ — тождественно истинное условие срабатывания.

Отметим, что при задании $\mathcal{O}$ никаких ограничений на $\mathcal{D}$, σ и ω не накладывается. Легко видеть, что при фиксированном n любая

реализация $\mathcal{R}$ вида (3) имеет размер $l = 3n$ и структуру

$$\mathcal{P}_{3j-2}\,[s_j\rangle_{\mathrm{n}}\,\mathcal{P}_{3j-1}\,[s_j\rangle_{\mathrm{r}}\,\mathcal{P}_{3j}\,[s_j\rangle_{\mathrm{c}}, \quad j = 1, \ldots, n,$$

причем

- $\mathcal{S}_r^{3j-1} = \{s_j\}$ и $\mathcal{S}_r^{3j-2} = \mathcal{S}_r^{3j} = \varnothing$;
- $\mathcal{S}_c^{3j-2} = \mathcal{S}_c^{3j-1}$ и $\mathcal{S}_c^{3j} = \mathcal{S}_c^{3j-1} \cup \{s_j\}$;
- $\mathcal{S}_c^{3j} = \mathcal{S}_c^{3j+1}$ при $j < n$.

Что и требовалось показать.

Рассмотрим следующую группу шаблонов WP2 и WP10, основанных на понятии нити управления. Шаблоны из данной группы либо описывают конструкции элементов управления, отвечающих за расщепление и соединение различных нитей управления, либо задают топологические ограничения на нить управления в рамках графовой структуры процесса.

Прежде чем непосредственно перейди к определению понятия нити управления, необходимо ввести понятия входящей ветви и функции времени срабатывания переходов.

ОПРЕДЕЛЕНИЕ 16 (входящая ветвь). Последовательность переходов β вида

$$(s_j, s_{j+1}) \in \mathcal{T}, \quad s_j \neq s_{j+1} \quad j = 1, \ldots, p, \tag{4}$$

назовем *входящей ветвью* для шага s_{p+1}. Число p назовем *длиной* входящей ветви β, а (s_p, s_{p+1}) — *заключительным переходом.*

Множество всех отличных от петель переходов процесса задает на множестве входящих ветвей отношение эквивалентности.

ОПРЕДЕЛЕНИЕ 17 (эквивалентность входящих ветвей). Будем говорить, что две входящие ветви β_1 и β_2 *эквивалентны*, и записывать $\beta_1 \sim \beta_2$, если β_1 и β_2 обладают одним заключительным переходом.

Для использования понятия нити управления необходимо фиксировать моменты срабатывания переходов при выполнении реализации процесса.

ОПРЕДЕЛЕНИЕ 18 (время срабатывания переходов). Каждой реализации процесса $\mathcal{R}$ вида (3) поставим в соответствие

функцию $\tau_{\mathcal{R}} : \mathcal{T} \longrightarrow \mathbb{N} \cup \{0\}$, которую будем интерпретировать как *время срабатывания переходов*. По определению положим

$$\tau_{\mathcal{R}}(t) \rightleftharpoons \begin{cases} n, & \text{если } t \notin \mathcal{T}_c^{n-1} \wedge t \in \mathcal{T}_c^n \wedge 2 \leqslant n \leqslant l; \\ 0, & \text{в противном случае.} \end{cases}$$

Функция $\tau_{\mathcal{R}}$ задана корректно. Изменение множества сработавших переходов может произойти только в результате срабатывания отношений вида $[s)_c$, $s \in \mathcal{S}$. В любой реализации процесса при фиксированном шаге s отношение $[s)_c$ может сработать не более одного раза. Следовательно, и каждый переход может присоединиться к сработавшим переходам не более одного раза.

ОПРЕДЕЛЕНИЕ 19 (нить управления). Входящую ветвь β вида (4) будем называть *нитью управления* реализации $\mathcal{R}$, если одновременно выполняются неравенства

$$0 < \tau_{\mathcal{R}}(s_j, s_{j+1}) < \tau_{\mathcal{R}}(s_{j+1}, s_{j+2}), \quad j = 1, \ldots, p-1.$$

Величину $\tau_{\mathcal{R}}(\beta) \rightleftharpoons \tau_{\mathcal{R}}(s_p, s_{p+1})$ назовем *временем срабатывания нити управления* β.

ПРИМЕР 2. Рассмотрим определение процесса $\mathcal{O}_{\text{п}}$ из примера 1 и две его реализации $\mathcal{R}_1$ и $\mathcal{R}_2$. Данные реализации имеют общую начальную часть. Срабатывание всех переходов процесса происходит при смене этапов от $\mathcal{P}_3$ к $\mathcal{P}_4$. Данные промежуточные этапы имеют соответственно номера 3 и 4. Поэтому $\tau_{\mathcal{R}_1} = \tau_{\mathcal{R}_2} = 4$. Таким образом, реализации $\mathcal{R}_1$ и $\mathcal{R}_2$ имеют две нити управления (s_1, s_2) и (s_1, s_3). Нить управления (s_1, s_2) является входящей ветвью для шага s_2, а нить управления (s_1, s_3) является входящей ветвью для шага s_3.

Подчеркнем важное различие между входящей ветвью и нитью управления. В то время как длина входящей ветви может быть сколь угодно большой, длина нити управления всегда ограничена.

ПРЕДЛОЖЕНИЕ 7 (оценка длины нити управления). *В любой реализации, соответствующей определению процесса* $\mathcal{O}$, *длина каждой нити управления не превосходит числа* $\operatorname{card}^2 \mathcal{S}$.

ДОКАЗАТЕЛЬСТВО. Из определения следует, что в нити управления все переходы различны. Поэтому длина нити управления не превосходит числа всех переходов процесса, которое,

очевидно, не больше чем число всевозможных пар шагов процесса. □

Приведенная оценка может служить утверждением о невозможности поддержки шаблона WP10 языком XPDL. Шаблон WP10 описывает конструкцию «произвольных» циклов. В данном шаблоне подразумевается, что процесс может иметь произвольную графовую структуру, и в нем может быть выделена специальная позиция. Пересекая данную позицию, нить управления порождает новую реализацию шага процесса. При этом, нити управления может соответствовать сколь угодно сложный путь в графе процесса. В качестве необходимого условия поддержки шаблона WP10 естественно выбрать существование определения процесса, реализации которого могут содержать нити управления произвольной длины. Данное условие в силу предложения 7 не может быть выполнено.

Формализуем конструкцию шаблона WP2, описывающего функционирование точки параллельного расщепления. Нить управления, пересекая данную точку, должна образовать целый набор параллельно выполняющихся нитей управления. При этом число вновь созданных нитей фиксировано.

ОПРЕДЕЛЕНИЕ 20 (точка параллельного расщепления). В определении процесса $\mathcal{O}$ шаг $s \in \mathcal{S}_{a,\,a} \setminus \mathcal{S}_f$ назовем *точкой параллельного расщепления*, если

- $(s, s) \notin \mathcal{T}$;
- $\operatorname{card} i(s) = 1$;
- $\operatorname{card} o(s) \geqslant 1$;
- $\lambda(t, \delta) = 1$, для всех $t \in o(s)$ и $\delta \in \mathcal{D}$.

ПРЕДЛОЖЕНИЕ 8 (поддержка шаблона WP2). *Предположим, что*

- s — *точка параллельного расщепления процесса* $\mathcal{O}$;
- $\mathcal{R}$ — *реализация процесса* $\mathcal{O}$ *вида* (3);
- $s \in \mathcal{S}_c^l$ — *шаг* s *завершает свое выполнение в реализации* $\mathcal{R}$;

- $t \in i(s)$ — *входящий в* s *переход.*

Тогда существует реализация $\mathcal{R}'$ *процесса* $\mathcal{O}$, *совпадающая с* $\mathcal{R}$ *исходными данными и результатом, в которой для любого исходящего перехода* $t' \in o(s)$ *последовательность* t, t' *является нитью управления.*

ДОКАЗАТЕЛЬСТВО. Условие $s \in \mathcal{S}_c^l$ означает, что реализация $\mathcal{R}$ должна содержать срабатывание отношения $[s\rangle_r$, которому, в свою очередь, должно предшествовать срабатывание отношения $[s\rangle_n$. Таким образом, для некоторых p, q, $1 \leqslant p < q < l$, выполняются условия $\mathcal{P}_p\,[s\rangle_n\,\mathcal{P}_{p+1}$ и $\mathcal{P}_q\,[s\rangle_r\,\mathcal{P}_{q+1}$. Условие $s \in_n \mathcal{P}_p$ является необходимым для срабатывания отношения $[s\rangle_n$. Поэтому, учитывая предположение $s \in \mathcal{S}_{a,a}$, получим включение $t \in \mathcal{T}_c^p$. Следовательно, справедлива оценка $0 < \tau_{\mathcal{R}}(t) \leqslant p$.

Далее, возможны два случая. В первом случае реализация $\mathcal{R}$ содержит срабатывание отношения $[s\rangle_c$. В данном случае в качестве искомой реализации $\mathcal{R}'$ будет выступать исходная реализация $\mathcal{R}$. Учитывая порядок срабатывания отношений, для некоторого h, $q < h < l$, имеем $\mathcal{P}_h\,[s\rangle_c\,\mathcal{P}_{h+1}$. При этом $\tau_{\mathcal{R}}(t') = h+1$ для каждого исходящего перехода $t' \in o(s)$. Таким образом, выполняются неравенства $0 < \tau_{\mathcal{R}}(t) \leqslant p < q < h < \tau_{\mathcal{R}}(t')$, и, следовательно, пара t, t' является нитью управления в реализации $\mathcal{R}$.

Во втором случае реализация $\mathcal{R}$ не содержит срабатывание отношения $[s\rangle_c$. По определению точки параллельного расщепления шаг $s \notin \mathcal{S}_f$ не является заключительным. Следовательно $s \neq s_{l-1}$ потому, что реализация должна всегда заканчиваться срабатыванием отношения завершения выполнения некоторого заключительного шага процесса. Поэтому $s \in \mathcal{S}_c^{l-1}$. Учитывая предположение относительно отсутствия срабатывания отношения $[s\rangle_c$, получим $o(s) \cap \{\mathcal{T}_c^{l-1} \cup \mathcal{T}_a^{l-1}\} = \varnothing$. Но тогда выполняется отношение $s \in_c \mathcal{P}_{l-1}$, и для некоторого промежуточного этапа $\mathcal{P}'_{l-1}$ получим $\mathcal{P}_{l-1}\,[s\rangle_c\,\mathcal{P}'_{l-1}$. Этапы $\mathcal{P}_{l-1}$ и $\mathcal{P}'_{l-1}$ отличаются только множествами сработавших переходов. У последнего оно равно $\mathcal{T}_c^{l-1} \cup o(s)$. Следовательно, $\mathcal{P}'_{l-1}\,[s_{l-1}\rangle_r\,\mathcal{P}_l$. Реализация $\mathcal{R}'$, состоящая из этапов $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_{l-1}, \mathcal{P}'_{l-1}, \mathcal{P}_l$, совпадает с $\mathcal{R}$ своими исходными данными и результатом. Кроме того, к $\mathcal{R}'$

применим ранее рассмотренный случай, что и завершает доказательство предложения.

Следующий шаблон WP7 реализует конструкцию точки синхронизирующего соединения. В реализации процесса точка синхронизирующего соединения может сработать (передать управление) только в том случае, когда не существует активной нити управления, которая может достигнуть данной точки. Рассматриваемый шаблон не поддерживается языком XPDL. Прежде, чем сформулировать необходимое условие поддержки шаблона WP7 с его последующем опровержением, введем понятие тестового процесса.

ОПРЕДЕЛЕНИЕ 21 (тестовый процесс для шаблона WP7). Определение процесса $\mathcal{O}$ назовем *тестовым процессом для шаблона* WP7, если для некоторых шагов $s_1, s_2, s_1', s_2' \in \mathcal{S}$ и для каждого состояния $\delta \in \mathcal{D}$ выполняются условия

- $\mathcal{S}_s = \{s_1\}$;
- $\mathcal{S}_f = \{s_2\}$;
- $o(s_1) = \{(s_1, s_1')\}$, $\lambda(s_1, s_1', \delta) = 0$;
- $i(s_2) = \{(s_2', s_2)\}$, $\lambda(s_2', s_2, \delta) = 1$.

В тестовом для шаблона WP7 процессе совокупность всех шагов, отличных от начального и заключительного, вместе со всеми переходами их соединяющими интерпретируется как возможная (предполагаемая) конструкция для точки синхронизирующего соединения. Тем самым рассматривается процесс, реализующий передачу управления от начального шага к заключительному, через точку синхронизирующего соединения. В качестве необходимого условия для поддержки шаблона WP7 естественно выбрать наличие у тестового процесса хотя бы одной реализации.

ПРЕДЛОЖЕНИЕ 9 (невозможность поддержки шаблона WP7). *Тестовый для шаблона* WP7 *процесс не имеет реализаций.*

ДОКАЗАТЕЛЬСТВО. Предположим, что тестовый для шаблона WP7 процесс $\mathcal{O}$ имеет реализацию. Так как $\mathcal{O}$ содержит единственный начальный шаг s_1, то любая реализация данного процесса должна начинаться с последовательного срабатывания

отношений $[s_1\rangle_n$, $[s_1\rangle_r$ и $[s_1\rangle_c$. В результате возникает промежуточный этап вида $\mathcal{P} = (\mathcal{O}; \varnothing, \{s_1\}, \varnothing, \{(s_1, s_1')\}; \delta)$. Этап $\mathcal{P}$ не является заключительным и отражает ситуацию, при которой шаг s_1 является выполненным, а переход (s_1, s_1') является не сработавшим.

При $\mathcal{P}$ не может сработать отношение смены состояний. Действительно, отношение запуска не может сработать потому, что не выполняется ни одно из необходимых для этого условий. Процесс $\mathcal{O}$ не содержит отличных от s_1 начальных шагов, а этап $\mathcal{P}$ не содержит сработавших переходов. Отношение завершения выполнения не может сработать, так как $\mathcal{P}$ не содержит выполняющихся шагов. Поскольку шаг s_1 при состоянии $\mathcal{P}$ уже передал свое управление, и $\mathcal{P}$ не содержит других выполненных шагов, то срабатывание отношения передачи управления также невозможно. Таким образом, предполагаемая реализация должна содержать тупиковый этап. Противоречие. □

Аналогично, при помощи использования подходящего тестового процесса может быть показана невозможность поддержки шаблона WP17. Данный шаблон реализует конструкцию перемежающейся параллельной маршрутизации. Рассматриваемая конструкция заключается в возможности выделить набор шагов процесса, которые, с одной стороны должны выполняться последовательно, а с другой стороны могут выполняться в произвольном порядке. Необходимым условием для поддержки данного шаблона является существование определений процессов, все шаги которых образуют перемежающуюся параллельную маршрутизацию.

ОПРЕДЕЛЕНИЕ 22 (тестовый процесс для шаблона WP17). Определение процесса $\mathcal{O}$ назовем *тестовым процессом для шаблона* WP17, если

- $\mathcal{S} = \mathcal{S}_S$ (возможность произвольного порядка выполнения шагов означает, что каждый шаг должен быть начальным);
- в любой реализации данного процесса $\mathcal{R}$ вида (3) выполняются условие $\operatorname{card} \mathcal{S}_r^i \leqslant 1$, $i = 1, \ldots, l$.

Число шагов будем называть *размером* тестового процесса.

ПРЕДЛОЖЕНИЕ 10 (невозможность поддержки шаблона WP17). *Не существует тестовых для шаблона* WP17 *процессов размера больше единицы.*

ДОКАЗАТЕЛЬСТВО. Допустим, что существует тестовый для шаблона WP17 процесс $\mathcal{O}$, у которого $\operatorname{card} \mathcal{S} \geqslant 2$. Тогда найдется пара различных шагов данного процесса $s_1, s_2 \in \mathcal{S}$, причем $s_1 \in \mathcal{S}_f$. Для любого начального этапа $\mathcal{P}_1$ существует реализация вида $\mathcal{P}_1 [s_1\rangle_n \mathcal{P}_2 [s_2\rangle_n \mathcal{P}_3 [s_1\rangle_r \mathcal{P}_4$. Рассматриваемая реализация заключается в последовательном запуске начальных шагов s_1 и s_2 и завершении выполнения заключительного шага s_2. При этом на этапе $\mathcal{P}_3$ имеется два выполняющихся шага, $\mathcal{S}_r^3 = \{s_1, s_2\}$. Данное обстоятельство противоречит определению тестового для шаблона WP17 процесса. □

Последним рассмотрим шаблон WP11, предполагающий возможность неявного завершения реализации процесса. Неявное завершение срабатывает в том случае, когда больше невозможно выполнение ни одного шага процесса. Данный шаблон не может напрямую поддерживаться языком XPDIL, так как в рамках формальной модели завершение реализации процесса может произойти только в результате выполнения заключительного шага.

Введем понятие тупикового этапа, на основании которого будет сформулировано утверждение о невозможности поддержки шаблона WP11 языком XPDIL.

ОПРЕДЕЛЕНИЕ 23 (тупиковый этап). Этап $\mathcal{P} \in \mathbb{P}(\mathcal{O}) \setminus \mathbb{F}(\mathcal{O})$ назовем *тупиковым*, если не существует другого этапа $\mathcal{P}' \in \mathbb{P}(\mathcal{O})$ такого, что $\mathcal{P} [\rangle \mathcal{P}'$.

ПРЕДЛОЖЕНИЕ 11 (невозможность поддержки шаблона WP11). *Существует процесс* $\mathcal{O}$, *у которого пересечение множества регулярных этапов и множества тупиковых этапов не пусто.*

ДОКАЗАТЕЛЬСТВО. В качестве искомого процесса возьмем процесс $\mathcal{O}$, состоящий только из начального шага s_1 и заключительного шага s_2. Множество переходов процесса $\mathcal{O}$ пусто. При любом начальном этапе может сработать только последовательность отношений $[s_1\rangle_n$ и $[s_1\rangle_r$, порождая регулярный промежуточный этап вида $\mathcal{P} = (\mathcal{O}; \varnothing, \{s_1\}, \varnothing, \varnothing; \delta)$. Данный этап не

является заключительным, и при нем не может сработать отношение смены состояний. Действительно, шаг s_1 не может передать управление, так как не имеет исходящих переходов. В то же время, неначальный шаг s_2 не может получить управление из-за отсутствия входящих в него переходов. □

Учитывая, что свойство тупиковости этапов выполнения реализаций процессов алгоритмически разрешимо, будем говорить о частичной поддержке шаблона WP11 языком XPDIL.

ПРЕДЛОЖЕНИЕ 12. *Свойство тупиковости этапов выполнения реализаций процессов алгоритмически разрешимо.*

ДОКАЗАТЕЛЬСТВО. Процедура проверки тупиковости этапа $\mathcal{P} \in \mathbb{P}(\mathcal{O})$ заключается в следующем. Тупиковый этап не может быть заключительным. Следовательно, сначала должно быть проверено, что конечные множества $\mathcal{S}_c$ и $\mathcal{S}_f$ не пересекаются. Далее, для каждого шага процесса $s \in \mathcal{S}$ необходимо убедиться в том, что одновременно не выполняются отношения $s \in_n \mathcal{P}$, $s \in_r \mathcal{P}$ и $s \in_c \mathcal{P}$. Учитывая конечность множества шагов и переходов процесса, данная проверка может быть эффективно осуществлена за конечное время. □

Суммируем в виде теоремы полученные результаты о выразительности языка XPDIL.

ТЕОРЕМА (выразительность языка XPDIL). *Среди шести протестированных шаблонов в формальной модели языка* XPDIL *напрямую поддерживаются два шаблона (шаблон последовательности* WP1 *и шаблон точки параллельного расщепления* WP2*), имеет частичную реализацию один шаблон (шаблон неявного завершения* WP11*) и не поддерживаются три шаблона (шаблон точки синхронизирующего соединения* WP7*, шаблон произвольных циклов* WP10 *и шаблон перемежающейся параллельной маршрутизации* WP17*).*

ДОКАЗАТЕЛЬСТВО. Утверждение теоремы является объединением доказанных ранее предложений 6–12.

Заключение

В настоящей работе была представлена формальная модель языка XPDIL, являющегося основой специализированной системы автоматизации базис процессов, предназначенной для

автоматизации сложных научных экспериментов. На основе формальной модели было произведено исследование выразительности языка XPDIL в рамках подхода, разработанного Ван-дер-Альстом и Хофстедом.

Отличительной чертой языка XPDIL является возможность в едином унифицированном виде представлять не только определения процессов, но и промежуточные этапы выполнения реализаций процессов, а также представлять сами реализации процессов целиком. Введенные семейства отношений вида $[s\rangle_n$, $[s\rangle_r$, $[s\rangle_c$, отвечающие за смену этапов выполнения реализаций процессов, позволяют дополнительно моделировать функционирование системы автоматизации бизнес-процессов, основанной на языке XPDIL. При этом оказывается, что выполнение реализации процесса в рамках формальной модели языка XPDIL не является детерминированной процедурой. Достаточно привести разобранный выше пример определения процесса $\Theta_\text{п}$, две реализации которого имеют одинаковые исходные данные и различные результаты. Подобная недетерминированность на практике возникает за счет внешних по отношению к системам автоматизации бизнес-процессов событий, влияющих на выполнение реализаций процессов. Внешние события формализованы в виде отличия между возможностью срабатывания шага процесса и самим срабатыванием. Возможность срабатывания шага процесса определяется отношениями $\in_n$, $\in_r$, $\in_c$, а срабатывание шага процесса определяется отношениями вида $[s\rangle_n$, $[s\rangle_r$, $[s\rangle_c$.

При исследовании выразительности выше были рассмотрены шесть наиболее представительных с точки зрения языка XPDIL шаблонов. Для каждого шаблона в терминах формальной модели языка XPDIL либо формулировалось и доказывалось утверждение об его поддержке, либо формулировалось необходимое условие его поддержки для последующего опровержения. При этом оказалось, что ряд шаблонов, таких как WP7 — точка синхронизирующего соединения и WP17 — перемежающаяся параллельная маршрутизация, не поддерживаются языком XPDIL. Данные шаблоны являются уникальными и на практике поддерживаются всего несколькими промышленными системами автоматизации бизнес-процесса. Шаблон WP17, например, под-

держивают только две системы автоматизации бизнес процессов — Cosa и Mobile. Таким образом, полученные отрицательные результаты в отношении некоторых шаблонов говорят скорее не о недостатках языка XPDIL, а об уникальности данных шаблонов. Результат о невозможности поддержки уникального шаблона необходимо интерпретировать как невозможность реализации его конструкции элементарными средствами. Поэтому в дальнейшем развитии языка XPDIL конструкции данных шаблонов должны добавляться в виде новых элементов управления.

Отсутствие поддержки языком XPDIL шаблона WP10 — конструкции «произвольных» циклов, также не является существенным ограничением. В языке XPDIL могут выделяться шаги процесса, для которых возможно многократное повторение. Таким образом, конструкция «простых» циклов поддерживается. Кроме того, уже в нынешней реализации языка XPDIL графовая структура процессов может быть сколь угодно топологически сложной. Для решения практических задач в настоящее время большинство шагов процессов требует не более одной реализации (шаг процесса, отличный от тела цикла, выполняется не более одного раза). Следовательно, отказываясь от данного ограничения, может быть получена реализация шаблона WP10 в языке XPDIL.

В заключение отметим, что проведенное исследование показывает, что язык XPDIL, с одной стороны, обладает необходимым минимумом выразительных средств, а с другой стороны, он не содержит принципиальных ограничений для расширения его выразительности.

Литература

1. ELIS C.A., NUTT G.J. Office Information Systems and Computer Science //ACM Computer Surveys. — 1980. — Vol. 12, № 1. — P. 27–60.

2. ВАСЕНИН В.А., АФОНИН С.А., КОРШУНОВ А.А. К созданию концепции интегрированной системы распределенных информационных ресурсов Московского государственного университета им. М.В.Ломоносова. М.: Изд во Моск. ун та, 2001. — 122 с.

3. БЫЛЯ О.И., ВАСИН Р.А., ДЕНИСОВ И.Ю., КОЗИЦЫН А.С., ШУНДЕЕВ А.С. К созданию информационно-экспертной системы по структурно-механическим свойствам материалов //Ломоносовские чтения, 2003. Тезисы докладов. — М.: Изд-во Моск. ун-та, 2003. — С. 112.

4. Workflow Management Coalition. Terminology and Glossary //Document WFMC-TC-1011. — Feb. 1999. — Issue 3.0. — 65 p.

5. Workflow Management Coalition. Workflow Process Definition Interchange — XML Process Definition Language //Document WFMC-TC-1025. — Oct. 2002. — Issue 1.0. — 87 p.

6. AALST W.M.P. van der . Patterns and XPDL: A critical Evaluation of the XML Process Definition Language //Queensland University of Technology, 2003. — Technical report FIT-TR-2003-06. — 30 p.

7. AALST W.M.P. van der , HOFSTEDE A.H.M. ter , KIEPUSZEWSKI B., BARROS A.P. Workflow patterns //Distributed and Parallel Databases. — 2003. — Vol. 14, № 1. — P. 5-51.

8. Workflow patterns page. http://www.workflowpatterns.com

9. ШУНДЕЕВ А.С. Управление автоматизированными бизнес-процессами на основе XML, Информационные технологии и программирование. — М.: Изд-во МГИУ, 2003. — Т. 1, № 6. — С. 31-44.

Поступила в редакцию
2 июля 2004 года