УДК 658.512

Н.Н. Войт

ПРИНЦИП «АНСАМБЛЬ» И АНАЛИТИЧЕСКАЯ МОДЕЛЬ ОРГАНИЗАЦИИ ПОТОКОВ ПРОЕКТНЫХ РАБОТ¹

Войт Николай Николаевич, кандидат технических наук, доцент, окончил факультет информационных систем и технологий Ульяновского государственного технического университета, доцент кафедры «Вычислительная техника» УлГТУ. Имеет более 200 научных статей в области интеллектуальных САПР, Case-, Cals-технологий. Область научных интересов: автоматизация проектирования потоков работ, формальные языки и грамматики, автоматы, контроль и анализ бизнес-процессов, интеллектуальные системы обработки workflows, онтологии, автоматизированные среды обучения. [e-mail: n.voit@ulstu.ru].

Аннотация

Целью фундаментального исследования автора является повышение эффективности обработки бизнес-процессов в автоматизированных системах (АС) в базе графических языков за счет сокращения временных затрат на их разработку и обработку. Целью также является повышение успешности обработки диаграмматических моделей потоков работ. Причем под успешностью понимается выполнение требований к ресурсным ограничениям, включая функциональность, финансовую составляющую и сроки исполнения. Также к цели относится повышение качества диаграмматических моделей с помощью контроля денотативных и сигнификативных семантических ошибок. Сам бизнес-процесс представлен в виде диаграмматической модели потоков работ, а под обработкой процессов понимается анализ, контроль, преобразование и интерпретация диаграмматических моделей потоков работ.

В работе содержится анализ существующих подходов к автоматизации проектирования потоков работ в условиях крупной проектной организации. Предложены описания разработки теоретических основ обработки диаграмматических моделей потоков работ в виде нового принципа организации потоков проектных работ в базе графических языков – «Ансамбль», новой аналитической модели организации потоков проектных работ с учетом оркестровки, хореографии. Разработанные принцип, модель интегрированы в методологию структурно-семантического подхода. Описан вычислительный эксперимент на базе разработанной модели, представляющий потоки проектных работ при конструкторской и технологической подготовках производства.

Настоящая статья является одной из последующих работ на тему методологии обработки потоков проектных работ в базе графических языков.

Ключевые слова: бизнес-процессы, графические языки, потоки работ, анализ, синтез.

doi: 10.35752/1991-2927-2020-3-61-124-137

THE ENSEMBLE PRINCIPLE AND ANALYTICAL MODEL OF PROJECT WORKFLOW ORGANIZATION

Nikolai Nikolaevich Voit, Candidate of Sciences in Engineering, Associate Professor; graduated from the Faculty of Information Systems and Technologies of Ulyanovsk State Technical University; Associate Professor at the Department of Computer Engineering of Ulyanovsk State Technical University; an author of more than 200 scientific articles in the field of intelligent CAD, Case-, Cals-Technologies; the area of his scientific interests relates to the workflow design automation, formal languages and grammars, automata, control and analysis of business processes, intelligent workflow processing systems, ontologies, automated learning environments. e-mail: n.voit@ulstu.ru.

Abstract

The purpose of the author's fundamental research is to increase the efficiency of processing business processes in automated systems (AS) in the basis of graphic languages by reducing the time spent on their development and processing. The goal is

¹ Исследование выполнено при финансовой поддержке РФФИ и Ульяновской области в рамках научного проекта № 18-47-730032.

also to improve the success of the processing of workflow diagrammatic models. Moreover, success is understood as meeting the requirements for resource constraints, including functionality, financial component and deadlines. Also, the goal includes improving the quality of diagrammatic models by controlling denotative and significative semantic errors. The business process itself is presented in the form of a diagrammatic workflow model, and process processing refers to the analysis, control, transformation and interpretation of workflow diagrammatic models.

The paper contains an analysis of existing approaches to automating the design of workflows in a large design organization. Descriptions of the development of the theoretical foundations for the processing of diagrammatic models of workflows are proposed in the form of a new principle of organizing the flows of design work in the basis of graphic languages – “Ensemble”, a new analytical model of organizing the flows of design work, taking into account orchestration, choreography. The developed principle and model are integrated into the methodology of the structural-semantic approach. A computational experiment based on the developed model is described, which represents the workflows of design work in the design preparation of production and technological preparation of production.

This article is one of the follow-up works on the methodology for processing design workflows in the basis of graphic languages.

Key words: business processes, graphic languages, workflows, analysis, synthesis.

ВВЕДЕНИЕ

Автоматизированные системы определены в ГОСТ 34.601-90 как организационно-технические системы, обеспечивающие выработку решений на основе автоматизации информационных процессов в различных сферах деятельности (управление, проектирование, производство и тому подобное) или их сочетаниях, а также в Международном стандарте IEEE 1471 как сложные системы, интенсивно использующие программное обеспечение. В теориях «Управление бизнес-процессами (Business Process Management)», теоретической информатике, теории автоматизации проектирования стадии создания подобных систем представлены потоками проектных работ проектировщиков, интенсивно использующих программные (компьютерные) средства разработки. Обработка таких потоков работ (бизнес-процессов) в парадигме сквозного цифрового проектирования содержит ключевые проектные процедуры: анализ и синтез, которые являются одними из направлений исследований и в значительной степени влияют на результат и успех проектирования. При этом проблемой успешности проектных решений в этих теориях занимаются более 30 лет. Такое внимание к проблеме вызвано высокой степенью выхода разработок (проектных решений) за рамки запланированных временных, финансовых и функциональных параметров. В существующих теориях выявлены причины и выработаны рекомендации по повышению успешности проектирования сложных автоматизированных систем, однако, по данным международной компании Standish Group, занимающейся исследованиями в области успешности разработок автоматизированных систем, в настоящее время лишь 40% разработок заканчиваются успешно.

В настоящей статье описаны принцип «Ансамбль» и аналитическая модель организации потоков проектных работ, а в последующих работах автор опишет новые грамматики, методы и средства обработки потоков проектных работ в базисе графических языков.

1 Анализ существующих подходов

Анализ современных работ по тематике позволил сравнить полученные результаты с имеющимися и определить место, вклад настоящей работы в решение проблемы. Рассматривая современную ситуацию в научном сообществе по направлениям визуальных языков и анализу диаграмматических моделей, можно сказать, что интерес к ним не спадает. Статьи [1–4] проводят сравнительные исследования таких известных языков, как BPMN, UML, IDEF, EPC и других для использования в самых разных областях. А некоторые дополнительно представляют свои, как, например, в работе [5]. Большое количество авторов трудится над анализом получаемых диаграмм.

Авторами из статьи [6] предлагается постройка из диаграммы ориентированного графа и его матрицы смежности. Определены правила, каждое из которых нацелено на нахождение определенных типов ошибок. Ошибки выделены синтаксические. Недостатком подхода является большое количество вычислений из-за последовательного применения множества сложных правил.

В статье [7] авторами рассматривается диаграмма классов UML. Они предлагают решение для анализа диаграмм на структурные ошибки. В основе него используются графы, а также шаблоны и антишаблоны. По заявлению авторов, после преобразований задача анализа легко решается с помощью переборных алгоритмов.

В статье [8] решена задача классификации ошибок на восьми базовых UML-диаграммах. Данная классификация позволяет отнести все ошибки, выявленные на диаграмме вариантов использования, диаграмме классов, диаграмме последовательности, диаграмме кооперации, диаграмме состояний, диаграмме деятельности и диаграмме компонентов к той или иной группе. Методы для анализа данных ошибок авторами не предлагаются.

В статье [9] основной проблемой выделяют то, что существующие формальные методы позволяют предпо-

ложить о наличии ошибок, но не позволяют определить их количество. Авторами определены основные ошибки, которые могут возникать при формировании структуры потока управления бизнес-процессом. Предложен коэффициент, позволяющий определить наличие и количество ошибок, а также оптимизационная задача, решение которой позволяет сформировать рекомендации по совершенствованию модели бизнес-процесса. Однако недостатком их метода является возможность анализа очень ограниченного количества типов синтаксических ошибок. Семантические ошибки в работе не рассматриваются. В разработанных методах тоже есть указание на место ошибок и их количество, а вычислительная сложность не уступает указанной в статье [9].

Ошибки, возникающие во время выполнения объединенного рабочего процесса, могут стоить огромных затрат. Как правило, проверка BPMN-модели ограничена синтаксическими проверками, а возможности проверки на ошибки исполнения во время проектирования ограничены. В статье [10] представлен метод для проверки правильности выполнения диаграммы взаимодействий BPMN 2.0. При моделировании совместной работы важно убедиться, что модель будет выполнена так, как задумано. Для анализа ошибок в статье вводятся концепции логических часов и логического времени, которые позволяют упорядочить различные исполняемые события, задачи, потоки сообщений и т. д. Каждый процесс распределенной системы имеет логическую метку времени, основанную на векторах меток времени. Процесс увеличивает свой временной вектор, когда происходит событие. Когда процесс отправляет сообщение, он включает в себя свой временной вектор с сообщением. Получив сообщение, получатель объединяет свои знания о времени с вектором времени в сообщении. Анализ векторов меток позволяет выявить проблемы, связанные с параллельным и последовательным порядком выполнения событий. Похожая концепция заложена в темпоральную автоматную грамматику, доработанная версия которой используется для анализа и контроля денотативной и сигнификативной семантики диаграмматических моделей гибридных динамических потоков проектных работ. Существенных преимуществ по сравнению с авторскими решениями не выявлено.

В статье [11] цель работы состоит в том, чтобы удалить из модели все элементы и структуры, которые являются правильными, но не значимыми для целей анализа, а затем изучить полученную модель. Авторы упрощают диаграмму для лучшей читабельности за счет сравнения с шаблонами и применения операций «вставить» и «удалить». Вычислительная сложность по их же подсчетам $O(n^2)$. Темпоральная автоматная грамматика при необходимости позволяет модифицировать диаграмму за меньшую вычислительную сложность.

Статья [12] призвана внести вклад в решение когнитивной проблемы (разработчик модели не в состоянии полностью передать свои знания о процессе в модели) путем анализа влияния путаницы во время моделирования на ошибки. Путаница может привести к ошибкам,

и поэтому ее следует избегать так же, как и реальных ошибок. Вклад статьи [12] заключается не в этом заключении как таковом, а в его подробном доказательстве (было проанализировано 31 588 операций и задокументировано 2 489 синтаксических проблем). Авторы предоставляют интересные знания о наличии различных типов сбивающих с толку конструкций и синтаксических ошибок и их потенциальных (причинных) связей. Однако используется сокращенный язык, искусственный случай и искусственные моделирующие модели, что ограничивает ценность исследования. Внимание в статье уделяется только синтаксису, хотя авторы и задумываются над семантическим анализом.

В статье [13] говорится о том, что в параллельных и распределенных системах становится все труднее обнаружить тупик и недетерминизм. Диаграммы активности UML представляют собой блок-схемы, которые моделируют последовательное и параллельное поведение. Авторы предлагают структуру, которая добавляет возможности проверки на указанные ошибки в инструмент моделирования UML и обеспечивает прозрачность формальной семантики для пользователей. Недостатком можно назвать более высокую вычислительную сложность для задачи, которая решена с помощью авторской темпоральной RVTI-грамматики.

В статье [14] предлагается усовершенствованный двухэтапный подход точного запроса, основанный на структуре графа. На этапе фильтрации составной индекс задачи, который состоит из метки, атрибута соединения и атрибута задачи, используется для получения моделей-кандидатов, что может значительно сократить количество моделей процессов, которые необходимо протестировать к определенному времени – алгоритм верификации. На этапе проверки предлагается новый тест на изоморфизм подграфа, основанный на коде задачи, чтобы уточнить набор моделей-кандидатов. Эксперименты проводятся на шести синтетических модельных наборах и двух реальных модельных наборах. Однако алгоритм имеет полиномиальную вычислительную сложность.

В статье [15] наиболее прочным фундаментом в теории потоков работ в настоящее время являются сети Петри. В анализе автор выделяет следующие ошибки: заблокированные задачи, тупик-«зависание», активный тупик (бесконечный цикл), выполнение задач после достижения конечной точки, наличие фишек в сети после завершения работы сети и другие. Также в [15] автор утверждает, что большинство современных языков моделирования потоков работ (BPMN, EPCs, FileNet и т.д.) построены на WF-сетях, подклассе сетей Петри. Однако, с учетом весьма широкого распространения инструментов работы именно с BPMN- и подобными диаграммами, для применения методов автора существует необходимость первоначальной трансляции исходных диаграмм в сеть Петри, что приводит к лишним затратам и источникам новых ошибок.

В работе [16] авторы расширяют алгоритм обобщенного LR на случай «грамматик с левым контекстом», ко-

торые дополняют контекстно-свободные грамматики специальными операторами для ссылки на левый контекст текущей подстроки, а также оператором соединения (как в конъюнктивных грамматиках) для объединения синтаксических условий. Все обычные компоненты алгоритма LR, такие как таблица синтаксического анализа, действия сдвига и уменьшения и т. д., расширены для обработки операторов контекста. Результирующий алгоритм применим к любой грамматике с левым контекстом, однако имеет такую же производительность в кубическом времени для наихудшего случая.

Также в работе [17] исследуются варианты операций объединения и конкатенации на формальных языках, в которых булева логика в определениях (то есть конъюнкция и дизъюнкция) заменяется операциями в двухэлементном поле GF (2) (конъюнкция и исключающее ИЛИ). Утверждается, что вычислительная сложность получается такая же, что и у обычных грамматик с объединением и конкатенацией: в частности, простой разбор во времени $O(n^3)$, что хуже, чем представляемая в проекте линейная производительность анализатора.

В статье [18] пишется, что алгоритм синтаксического анализа LL допускает произвольные контекстно-свободные грамматики и достигает хорошей производительности, но не может обрабатывать грамматики EBNF. Основным вкладом этой статьи является модификация алгоритма GLL, который может обрабатывать грамматики в форме, тесно связанной с EBNF (расширенной контекстно-свободной грамматикой). Как видно из работы, быстродействие увеличивается, однако все равно имеет степенной рост с приростом элементов анализируемой цепочки.

2 ОРКЕСТРОВКА, ХОРЕОГРАФИИ В САПР И ПРИНЦИП «АНСАМБЛЬ»

Технологии обмена сообщениями, благодаря которым несколько независимых участников-организаций

стремятся достичь желаемого состояния, называются «хореографией», а взаимодействие их отделов (служб) называется «оркестровкой». Специальные методологии и стандарты в САПР, такие как BPEL4WS, XLANG, WSFL, разработаны для оркестровки с целью описания бизнес-модели. Настоящие методики и инструменты разрабатываются с участием крупнейших поставщиков, таких как IBM, Microsoft, Oracle и BEA Systems и др. Оркестровка относится к определению бизнес-процесса, который может взаимодействовать с внешними и внутренними службами. Взаимодействия, возникающие на основе обмена сообщениями, включают бизнес-логику и порядок выполнения задач. Они могут выходить за рамки приложений и организаций, определяя долгосрочную, транзакционную, многошаговую бизнес-модель. Оркестровка – это всегда контроль с точки зрения одного участника в процессе. Хореография позволяет каждому участнику описать свою часть взаимодействия. При использовании хореографии отслеживаются последовательности сообщений между несколькими участниками и источниками.

Динамика потоков работ, представленных диаграмматическими моделями, рассматривается в двух аспектах: оркестровка и экземпляры хореографии, что в совокупности составляет ансамбль. Причем появившиеся экземпляры хореографии будут связаны с ранее спроектированными диаграммами. Например, организация 1 содержит BPMN, IDEF0, UML AD, eEPC, UML class диаграмм в оркестровке, но в ансамбле участвуют только BPMN, IDEF0, UML AD. Организация 2 содержит Java, C#, Timed automaton, Petri nets в оркестровке, но в ансамбле участвуют только Java и Timed automaton.

Рассмотрим оркестровку IDEF0 и UML AD диаграмм в ансамбле (рис. 1). IDEF0-диаграммы хорошо себя зарекомендовали как функциональный инструмент, учитывающий связи типа «Вход», «Выход», «Управление», «Механизмы». UML AD-диаграммы – как инструмент разработки программных средств в части алгоритмов. В тех случаях, когда нет необходимости учитывать управ-

Рис. 1. Ансамбль разнотипных диаграмм и гибридной оркестровки

ление и механизмы в IDEF0, а необходимо лишь отслеживать потоки.

Новый принцип предложенного структурно-семантического подхода, названный как «Ансамбль», заключается в композиции гибридных (гетерогенных) типов, распределенных во времени (динамических) и в пространстве потоков проектных работ в базисе графических языков, описывающих проектную деятельность крупного проектного предприятия, а именно конструкторских, технологических процессов подготовки производства, хорошо выраженных в оркестровке и хореографии; подчеркивает мощностные и выразительные возможности подхода при контроле, анализе, синтезе, преобразовании и интерпретации диаграмматических моделей потоков проектных работ в базисе графических языков в проектировании сложных автоматизированных технических систем, интенсивно использующих время; является фундаментальным в разработанных подходе, методах, моделях, алгоритмах и средствах.

3 АНАЛИТИЧЕСКАЯ МОДЕЛЬ ОРГАНИЗАЦИИ ПОТОКОВ ПРОЕКТНЫХ РАБОТ С УЧЕТОМ ОРКЕСТРОВКИ И ХОРЕОГРАФИИ

Композиция моделей хореографии, оркестровки в крупных проектно-производственных предприятиях представлена разработанной аналитической моделью CM и имеет следующий вид [19]:

$$CM = \{(Organization_i, Orchestration_i), \\ Choreography, Ensemble\},$$

где пара $(Organization_n, Orchestration_n)$ определяют внутренние потоки проектных работ организации, $i \in N$. Связь-метка n обеспечивает однозначную связь организации с ее оркестровкой. Хореография определена на всем множестве пар $(Organization_n, Orchestration_n)$ и является глобальной в системе, т. е. доступна из любой $Organization_i$ и $Orchestration_i$. $Organization_i = \{id, name\}$ является множеством, содержащим порядковый номер организации id и ее наименование $name$, где каждая проектно-промышленная организация идентифицируется по id , причем $id \in N$. Пусть $AllName$ является множеством всех наименований организаций, следовательно, $name \in AllName$.

Пусть $Variable$ является множеством переменных x, y, z . Обозначим $\tilde{x}$ кортеж переменных $\langle x_1, x_2, \dots, x_n \rangle$. Пусть $Command_{name}$ является множеством имен команд. $Command_{direction} = \{ow, rr\}$ является множеством типов команд, где ow обозначает односторонний обмен, тогда как rr обмен вида запрос-ответ. Сама команда описывается именем и типом:

$$command = \\ = \{(title, direction) \mid title \in Command_{name}, \\ direction \in Command_{direction}\},$$

является множеством команд (поток проектных работ), где каждая команда имеет уникальное имя.

Состояния системы переходов аналитической модели композиции хореографии и оркестровки определяется значениями переменных и вычисляется функцией обмена $Function_{rel} : Variable \rightarrow Value$. Причем область значения этой функции вычисляется по формуле Ψ и равна $v \in Value \cup \{\emptyset\}$. $Function_s(x) = \emptyset$ означает, что переменная x не задана. Запись $S[v/x]$ означает, что система в состоянии S имеет переменную x со значением v (следуя аналогии, можно сказать, что запись $S[\tilde{v}/\tilde{x}]$ означает, что кортеж переменных $\langle x_1, x_2, \dots, x_n \rangle$ имеет значения $\langle v_1, v_2, \dots, v_n \rangle$ соответственно):

$$S[v/x] = S', \\ S'(x') = \begin{cases} v, & \text{если } x' = x, \\ S(x'), & \text{в любом другом случае.} \end{cases}$$

Переменные могут принимать ограниченное множество значений. Бэкуса-Наура форма логических условий описания ограничений переменных позволит задать начальное состояние системы:

$$\phi ::= x \leq expression \mid expression \leq x \mid \neg \phi \mid \phi \wedge \phi,$$

где $expression$ является выражением, содержит переменные и принимает значение v в состоянии S . Запись $expression_s \rightarrow v$ означает, что выражение $expression$ вычисляется и принимает значение v в состоянии S . Как правило, для обозначения того, что условие ϕ выполняется в состоянии S , используется запись $S \vdash \phi$. Правила для ϕ определяются так:

1. $S(x) = \emptyset \Rightarrow S \vdash (x \not\leq \emptyset \wedge \emptyset \not\leq x)$,
2. $expression_s \rightarrow v, S(x) \not\leq v \Rightarrow S \vdash x \not\leq expression$,
3. $expression_s \rightarrow v, v \not\leq S(x) \Rightarrow S \vdash expression \not\leq x$,
4. $S \vdash \phi' \wedge S \vdash \phi'' \Rightarrow S \vdash \phi' \wedge \phi''$,
5. $\neg(S \vdash \phi) \Rightarrow S \vdash \neg \phi$.

Обмен между хореографией и оркестровкой определяется формальной записью следующим образом:

$$Interaction ::= 0 \mid \theta \mid Interaction_{in}; Interaction_{out} \mid \\ ::= Interaction_{in} \parallel Interaction_{out} \left[\sum_{i \in N}^+ \theta_i; Interaction_i \right] \\ ::= \sum_{i \in N}^{\oplus} \phi_i ? \theta_i; Interaction_i \\ \theta ::= (Choreography ::= \rho_A,$$

$$Orchestration_i ::= \left\{ (workflow, state)_{id_i}, \right. \\ \left. title, \tilde{x}, \tilde{y}, direction \right\} \mid x ::= expression,$$

где N обозначает натуральный ряд чисел, θ является потоком проектных работ обмена. Запись

$::= (\rho_A, (workflow, state)_{id_i}, title, \tilde{x}, \tilde{y}, direction)$
 означает, что обмен между хореографом ρ_A и оркестратором $(workflow, state)_{id_i}$ завершился успешно. $title$ обозначает имя команды обмена $(title, direction) \in Command$. Кортежи переменных $\tilde{x}, \tilde{y}$ нужны при отсылке и приеме потока проектных работ обмена соответственно. $direction$ задает отправку ($\uparrow$) и прием ($\downarrow$) потоков проектных работ. $x ::= expression$ означает, что вычисленное значение $expression$ присвоено переменной x .

Потоки проектных работ обмена могут принимать значение 0, базового оператора θ , иметь вид последовательности $(I; I)$, параллельности $(I \parallel I)$, выбора: безусловный (неограниченный) выбор $\sum_{i \in N}^+ \theta_i; I_i$, условный (ограниченный) выбор $\sum_{i \in N}^{\oplus} \phi_i ? \theta_i; I_i$. Неограниченный выбор выполняется независимо от состояния системы CM , описанного условием φ .

Определим действия при обмене потоков проектных работ как $exchange$ и представим его как

$$exchange_I = \{\lambda | \lambda = \left(\rho_A, (workflow, state)_{id_i}, title, \tilde{v}, direction \right) \cup \{time_{clock}\},$$

где λ является параметром обмена, $time_{clock}$ представляет время. Запись вида $(I, S) \xrightarrow{exchange} (I', S')$ означает, что произошло действие $exchange$, результатом которого стал обмен, и система CM из состояния (I, S) перешла в (I', S') .

Пусть Θ является множеством всех возможных состояний системы CM . Операционная семантика может быть представлена как $\rightarrow \Theta \times exchange \times \Theta$. Запишем в таблице 1 возможные обмены в модели композиции хореографии и оркестровки CM , учитывая $\delta \subseteq command$.

Таблица 1

Описание аксиом и правил операционной семантики обмена в модели CM

Аксиомы

Обмен № 1 – Отправка

$$\begin{aligned} & \left[\left(choreography, (workflow, state)_{id_i}, command, \tilde{x}, \tilde{y}, \uparrow \right), S \right] \xrightarrow{\lambda} (0, S[\tilde{\delta}/\tilde{y}]), \lambda \\ & = \left(choreography, (workflow, state)_{id_i}, command, \tilde{\delta}, \uparrow \right), \\ & \tilde{\delta} = S(\tilde{x}) \end{aligned}$$

Обмен № 2 – Прием

$$\begin{aligned} & \left[\left(choreography, (workflow, state)_{id_i}, command, \tilde{x}, \tilde{y}, \downarrow \right), S \right] \xrightarrow{\lambda} (0, S[\tilde{\delta}/\tilde{x}]), \lambda \\ & = \left(choreography, (workflow, state)_{id_i}, command, \tilde{\delta}, \downarrow \right), \\ & \tilde{\delta} = S(\tilde{y}) \end{aligned}$$

Правила

Обмен № 3 – Назначение

$$\frac{expression_s \rightarrow v}{(x ::= expression, S) \xrightarrow{\tau} (0, S[v/x])}$$

Обмен № 4 – Последовательность

$$\frac{(I, S) \xrightarrow{v} (I', S')}{(I; D, S) \xrightarrow{v} (I'; D, S')}$$

Обмен № 5 – Параллельность

$$\frac{(I, S) \xrightarrow{v} (I', S')}{(I \parallel D, S) \xrightarrow{v} (I' \parallel D, S')}$$

Обмен № 6 – Конгруэнтность

$$\frac{I \equiv I', (I, S) \xrightarrow{v} (D, S'), D \equiv D'}{(I', S) \xrightarrow{v} (D', S')}$$

Обмен № 7 – Выбор 1 (безусловный)

$$\frac{(\theta_i; I_i, S) \xrightarrow{v} (I'_i, S'), i \in N}{\left(\sum_{i \in N}^+ \theta_i; I_i, S\right) \xrightarrow{v} (I'_i, S')}$$

Обмен № 8 – Выбор 2 (условный)

$$\frac{S \vdash \phi_i, (\theta_i; S, t) \xrightarrow{v} (0, S'), S \not\vdash \phi_j, j \in N, j < i}{\left(\sum_{i \in N}^{\oplus} \phi_i ? \theta_i; I_i, S\right) \xrightarrow{v} (I_i, S')}$$

Структурная конгруэнтность

$$\begin{aligned} 0; I &\equiv I \\ I \parallel 0 &\equiv I \\ I \parallel D &\equiv D \parallel I \\ (I \parallel D) \parallel F &\equiv I \parallel (D \parallel F) \end{aligned}$$

Структурная конгруэнтность задает эквиваленты обмена причем $0; I$ является моноидом, в котором 0 является пустым множеством, кроме этого, означает конец обмена, после чего может начаться следующий обмен. Обмены № 1, № 2 описывают взаимодействие, которое являются запросом или ответом в зависимости от значения *direction*. При выполнении запроса (*direction* = $\uparrow$) информация, содержащаяся в переменных кортежа $\tilde{x}$ отправителя choreography, передается командой *command* в переменные кортежа $\tilde{y}$ получателю $(workflow, state)_{id_i}$. При приеме ответа (*direction* = $\downarrow$) информация, содержащаяся в переменных кортежа $\tilde{y}$ получателя $(workflow, state)_{id_i}$, передается командой *command* в переменные кортежа $\tilde{x}$ отправителю choreography. Обмен № 3 – Назначение обеспечивает присваивание результата выражения *expression* переменной x , вычисленное в состоянии S , обновляя состояние хореографии. Правила последовательности, параллельности и конгруэнтности определяют порядок обмена и эквивалентность соответственно. Правило выбора 1-безусловный обеспечивает дизъюнкцию выполнения обмена $(\theta_i; I_i, S) \xrightarrow{v} (\theta'_i, S')$. Правило выбора 2-условный имеет дело с условием ϕ_i , по которому выполняется строгая дизъюнкция обмена $(\theta_i; S) \xrightarrow{v} (\theta, S')$.

Ensemble является схемой интероперабельности (ансамблем) гибридных динамических диа-

грамматических моделей потоков проектных работ, определяющая множество функций обмена $Function_{rel}$ в хореографии и оркестровке. При этом $Ensemble = \{Interaction \rightarrow Map\}$ определяет, когда какие оркестраторы и хореографы (роли) будут задействованы в потоке проектных работ. Сценарий взаимодействия хореографов и оркестраторов задается в *Map* кортежем имен вызовов соответствующих команд обмена $Map = \langle (title_1, \tau_1), (title_2, \tau_2), \dots, (title_n, \tau_n) \rangle$, ранжированных по времени τ_i .

4 ОПИСАНИЕ ВЫЧИСЛИТЕЛЬНОГО ЭКСПЕРИМЕНТА

Конструкторская и технологическая подготовка производства на рисунке 2 содержит укрупненное описание двух типовых проектных процессов: конструкторской подготовки производства (КПП) и технологической подготовки производства (ТПП). Согласно методологии IBM RUP [20], структурно-функциональная модель описания проектных процессов крупного проектно-производственного предприятия должна содержать не более 11 уровней, что считается «хорошей» организацией труда. В нашем случае выделены укрупненные уровни, и их всего пять.

КПП декомпозируется на свои составные элементы, такие, как составление технического задания (Разработка ТЗ), составление конструкторской документации (Разработка КД), а также, в случае внесения изменений в техническую документацию, составление извещения об изменении (ИИ) в КД (Разработка ИИ) с последую-

щим утверждением его со стороны держателя документации на изделие. На четвертом уровне модели рисунку 2 содержатся потоки проектных работ, такие, как начало выполнения обозначенных задач технического задания (Инициализация работ), сам процесс проектирования КД (Разработка), далее следуют этап сопровождения этой документации в названные структурные подразделения предприятия с целью обсуждения (Согласование), подписание полученной КД (Утверждение), постановка на учет утвержденной документации (Сдача в архив).

Завершающий уровень представленной модели содержит конкретные действия в виде перечня потоков проектных работ и операций для решения поставленной на уровне четыре задачи. Инициализация работ содержит три потока проектных работ, а именно: 1) Анализ требования, 2) Определение исполнителей и 3) Определение сроков, и является стартовым укрупненным потоком проектных работ. Разработка содержит пять потоков проектных работ: 1) Анализ требования, 2) Определение инструментов разработки, 3) Проектирование, 4) Подбор/покупка материалов, номенклатуры и типовых решений, 5) Формирование комплекта документов, является базовым и важным в автоматизации проектирования сложных автоматизированных си-

стем. Завершающими этапами являются этап Согласования, который включает по два потока проектных работ, а именно: 1) Анализ конструктива, 2) Подписание; этап Сдачи в архив, состоящий из трех потоков проектных работ: 1) Проверка комплекта документации, 2) Постановка на учет/проведение изменений, 3) Абонентский учет.

ТПП декомпозируется на свои составные элементы такие, как составление маршрутной карты изделия (Разработка маршрута), составление управляющей программы для станков с ЧПУ и составление технологического процесса (Разработка ТП), в случае необходимости, конструирование специальной технологической оснастки в роли вспомогательного узлов, блоков, стоек и т. п. (Разработка СТО). Разработка ТП содержит три потока проектных работ, а именно: 1) Нормирование основных материалов, 2) Разработка ТП, 3) Нормирование трудовых норм.

Согласно рисунку 2 разработан экземпляр аналитической модели *СМ* согласования конструкторской документации при проектировании сложных автоматизированных систем для типового крупного проектно-производственного предприятия. Ансамбль аналитической модели *СМ* определяется следующим образом.

Рис. 2. Гибридные динамические потоки проектных работ «Конструкторская и технологическая подготовки производства»

$$\begin{aligned}
semble &= \left(\begin{aligned} &(Done\ 1,3), (Done\ 5,3), (Done\ 2,3), \left(\sum_{i=0}^1 (Done\ 6,3)?\uparrow 1;\uparrow 0 \right), \\ &\downarrow 0(Failed\ Done\ 1,0), \\ &(Done\ 4,3), (Done\ 3,3), \downarrow 1\left(\sum_{i=1}^1 (Successfully\ Done\ 1,0)?\uparrow 1;\uparrow 0 \right), \\ &\downarrow 0(Failed\ Done\ 2,0), \downarrow w(Done\ 4,3), (Done\ 3,3) \blacksquare \downarrow 1 \\ &\left(\sum_{i=0}^1 (Design\ and\ Laboratory, 0)?\uparrow 1;\uparrow 0 \right), \\ &\downarrow 0(Design\ and\ Laboratory\ Failed\ Done, 0), \uparrow w, \\ &\downarrow 1\left(\sum_{i=0}^1 (Design\ and\ Laboratory\ Successfully\ Done, 0)?\uparrow 1;\uparrow 0 \right), \\ &\downarrow 0(Approved\ Failed, 0) \uparrow w, \\ &\downarrow 1\left(\sum_{i=0}^1 (Approved\ Successfully, 0)?\uparrow 1;\uparrow 0 \right), \downarrow 0(OSN\ Failed, 0) \uparrow w, \\ &(OSN\ Successfully, 0) \blacksquare \end{aligned} \right) \\
Design\ and\ Laboratory &= \left(\begin{aligned} &\left(\sum_{i=0}^1 (Successfully\ Done\ 2,0)?\uparrow 2;\uparrow 1 \right), \downarrow 2(Design\ Done, 0) \uparrow w, \\ &\downarrow 1(Laboratory\ Done, 0), \downarrow w\left(\sum_{i=0}^1 (Exit\ from\ a\ Route, 0)?\uparrow 1;\uparrow 0 \right), \\ &\downarrow 0(Exit\ from\ a\ Route\ Failed\ Done, 0) \blacksquare, \\ &\downarrow 1(Exit\ from\ a\ Route\ Successfully\ Done) \blacksquare \end{aligned} \right) \\
Design\ Done &= \left(\begin{aligned} &(Design\ Done, 3), (NKS\ Done, 3), (Failed\ Done, 0), \\ &(NKS\ Successfully\ Done, 0), (Fork\ 'AND',), (T.Control\ Done, 0), \\ &(OVK\ Done, 0), (OPZM\ Done, 0), (Join\ 'AND'), \\ &(Head\ of\ a\ topic\ Failed\ Done, 0), \\ &(Head\ of\ a\ topic\ Successfully\ Done, 0), \\ &(Head\ of\ a\ sector\ Failed\ Done, 0), \\ &(Head\ of\ a\ sector\ Successfully\ Done, 0) \end{aligned} \right) \\
Laboratory\ Done &= \left(\begin{aligned} &(Laboratory\ Done, 5), (Technical\ sector\ Done, 0), \\ &(Fork\ 'AND', 0), (Laboratory\ OVK\ Done, 0), (Laboratory\ OMO\ Done, 0), \\ &(Join\ 'AND'), (OVK\ OPZM\ OMO\ Failed\ Done, 0), (for\ Remarks, 0), \\ &(OVK\ OPZM\ OMO\ Successfully\ Done, 0), (DM\ Failed\ Done, 0), \\ &(Department\ manager\ Checking\ Successfully\ Done, 0), \\ &(head\ of\ a\ topic\ Failed\ Done, 0), (Head\ of\ a\ topic\ Successfully\ Done, 0) \end{aligned} \right)
\end{aligned}$$

Начальные значения переменных модели CM имеют следующие значения.

$$X = \langle problem\ definition\ Control\ Centre ::= 200'1 \rangle$$

Граф-схема работ (ГСР) данного экземпляра аналитической модели представлена на рисунке 3.

Рис. 3. Гибридные динамические потоки проектных работ аналитической модели организации потоков работ на примере проектной процедуры согласования КД представлены в диаграмматическом виде

Результаты вычислительного эксперимента ГСР на примере решения задачи согласования КД (рис. 3) получены в CPN Tools и представлены в таблице 2. Критериями анализа были живость (Liveness), тупики (Dead Markings), 'справедливость' (Fairness), ограниченность (Boundedness). Полученные результаты подчеркивают

мощность и выразительность настоящей аналитической модели при решении задач анализа потоков проектных работ. Расшифровка критериев анализа таблицы 2 (Liveness, Dead Markings, Fairness, Boundedness и др.) представлена в документации на CPN Tools [21].

Таблица 2

Подробные результаты вычислительного эксперимента

Критерий		Значение
State Space		
Nodes:		41886
Arcs:		153698
Secs:		300
Status:		Partia
Scc Graph		
Nodes:		19805
Arcs:		94783
Secs:		4
Boundedness Properties		
Best Integer Bounds		
	Upper	Lower
Agreed_upon_document'agree 1	5	0
Agreed_upon_document'checking 1	9	0
Agreed_upon_document'development 1	27	0
Agreed_upon_document'problem_definition_Control_Centre 1	200	173
Agreed_upon_document'quire 1	13	0
Agreed_upon_document'remarks 1	5	0
Agreed_upon_document'rework 1	4	0
Agreed_upon_document'route 1	6	0
Design_Done'design_engineer 1	0	0
Laboratory_Done'Technical_sector 1	3	0
Successfully_Done'Select_a_document_type 1	4	0
Best Upper Multi-set Bounds		
Agreed_upon_document'agree 1	4`0++	
	5`1	
Agreed_upon_document'checking 1	9`1	
Agreed_upon_document'development 1	27`1	
Agreed_upon_document'problem_definition_Control_Centre 1	200`1	
Agreed_upon_document'quire 1	13`1	
Agreed_upon_document'remarks 1	5`1	
Agreed_upon_document'rework 1	4`1	
Agreed_upon_document'route 1	6`0++	
	5`1	
Design_Done'design_engineer 1	empty	
Laboratory_Done'Technical_sector 1	3`1	

Продолжение табл. 2

Successfully_Done'Select_a_document_type 1	4`0++
	4`1
Best Lower Multi-set Bounds	
Agreed_upon_document'agree 1	empty
Agreed_upon_document'checking 1	empty
Agreed_upon_document'development 1	empty
Agreed_upon_document'problem_definition_Control_Centre 1	173`1
Agreed_upon_document'quire 1	empty
Agreed_upon_document'remarks 1	empty
Agreed_upon_document'rework 1	empty
Agreed_upon_document'route 1	empty
Design_Done'design_engineer 1	empty
Laboratory_Done'Technical_sector 1	empty
Successfully_Done'Select_a_document_type 1	empty
Home Properties	
Home Markings	None
Liveness Properties	
Dead Markings	9579 [41886,41885,41884,41883,41882,...]
Dead Transition Instances	Design_Done'Design_Done 1
Live Transition Instances	None
Fairness Properties	
FImpartial Transition Instances	Agreed_upon_document'Done2 1 Agreed_upon_document'Done3 1 Agreed_upon_document'Done4 1 Agreed_upon_document'Done5 1 Agreed_upon_document'Done6 1
Fair Transition Instances	Design_Done'Design_Done 1
Just Transition Instances	None
Transition Instances with No Fairness	Agreed_upon_document'Done1 1 Agreed_upon_document'Failed_Done1 1 Agreed_upon_document'Failed_Done2 1 Agreed_upon_document'Successfully_Done1 1 Laboratory_Done'Laboratory_Done 1 Successfully_Done'Successfully_Done2 1

ЗАКЛЮЧЕНИЕ

Автором предложены новые принцип «Ансамбль», аналитическая модель организации потоков проектных работ с учетом хореографии и оркестровки, обеспечивающие повышение качества организации потоков проектных работ в САПР и автоматизированных системах технологической подготовки производства (АСТПП), а также описан вычислительный эксперимент, демонстрирующий качество разработанной аналитической модели. Модель отличается от существующих аналогов тем, что учитывает гибридный характер потоков проектных работ, а также распределенные во времени (дина-

мический характер), распределенные в пространстве потоки проектных работ в базе графических языков.

Предложенные принцип «Ансамбль» и аналитическая модель организации потоков проектных работ полностью интегрированы в компьютерную систему управления потоками проектных работ в АО «Ульяновский механический завод» в рамках выполнения хоздоговора НИР № Д31/31 «Разработка и исследование новых инструментальных средств повышения эффективности подготовки специалистов управления информационных технологий (УИТ), экспериментального программно-информационного обеспечения

по интеллектуальной базе опыта и управлению динамическими потоками проектных работ предприятия» (№ гос. рег. АААА-А17-117031410037-2, 2017 г.).

СПИСОК ЛИТЕРАТУРЫ

1. Сапожкова Т.Е. Сравнительный анализ подходов к моделированию бизнес-процессов // Прикладная информатика. 2012. №. 1 (37). С 14–19.
2. Суюнова Г.Б., Гайворонская Н.А., Половинко Е.В. Методы оптимизации экономических и социальных систем с использованием технологий моделирования бизнес-процессов // Моделирование, оптимизация и информационные технологии. 2018. Т. 6, №. 1. С. 176–184.
3. Макурков А.С., Шибанов С.В. Нотации для конструирования активных правил // Молодежь и наука: актуальные проблемы фундаментальных и прикладных исследований. 2019. С. 367–370.
4. Виролайнен А.В. Моделирование бизнес-процессов // Лучшая научная статья 2019 : сб. ст. XXIV Международ. науч.-исслед. конкурса. Пенза : МЦНС «Наука и Просвещение». 2019. С. 82–86.
5. Чернышов А.С. Анализ методов моделирования автоматизированных систем // Информационные технологии и автоматизация управления : сб. тр. конф. (Омск, 15–16 мая 2019 г.). 2019. С. 363–370.
6. Копп А.М., Орловский Д.Л. Подход к анализу и оптимизации моделей бизнес-процессов в нотации BPMN // Радиоелектроніка, інформатика, управління. 2018. №. 2 (45). С. 108–116.
7. Сергиевский М.В., Кирпичникова К.К. Валидация и оптимизация диаграмм классов UML // Cloud of science. 2018. Т. 5, № 2. С. 367–378.
8. Богдан И., Задорожний А. Классификация ошибок на UML-диаграммах, возникающих в ходе разработки IT-проектов // Технічні науки та технології. 2018. №. 1. С. 68–78.
9. Копп А.М., Орловский Д.Л. Аналіз та оптимізація моделей бізнес-процесів у нотациях ерс та bpmn // Технічні науки та технології. 2018. №. 4 (14). С. 145–152.
10. Design Time Validation for the Correct Execution of BPMN Collaborations / J. Anseeuw et al. // CLOSER (1). 2016. pp. 49–58.
11. A Pattern Based Method for Simplifying a BPMN Process Model / M. Ramos-Merino et al. // Applied Sciences. 2019. Vol. 9, № 11. P. 2322.
12. Claes J., Vandecaveye G. The impact of confusion on syntax errors in simple sequence flow models in BPMN // International Conference on Advanced Information Systems Engineering. Springer, Cham, 2019. pp. 5–16.
13. Lima L., Tavares A., Nogueira S.C. A framework for verifying deadlock and nondeterminism in UML activity diagrams based on CSP // Science of Computer Programming. 2020. P. 102497.
14. Huang H., Peng R., Feng Z. Efficient and exact query of large process model repositories in cloud workflow systems // IEEE Transactions on Services Computing. 2015. Vol. 11, №. 5. pp. 821–832.
15. Van der Aalst W.M. Everything You Always Wanted to Know About Petri Nets, but Were Afraid to Ask // International Conference on Business Process Management. Springer, Cham, 2019. pp. 3–9.
16. Barash M., Okhotin A. Generalized LR parsing algorithm for grammars with one-sided contexts // Theory of Computing Systems. 2017. Vol. 61, № 2. pp. 581–605.
17. Formal languages over GF (2) / E. Bakinova et al. // International Conference on Language and Automata Theory and Applications. Springer, Cham, 2018. pp. 68–79.
18. Gorokhov A., Grigorev S. Extended Context-Free Grammars Parsing with Generalized LL // International Conference on Tools and Methods for Program Analysis. Springer, Cham, 2017. pp. 24–37.
19. Анализ конструкторско-технологических потоков работ в условиях крупного радиотехнического предприятия / А.Н. Афанасьев, Н.Н. Войт, М.Е. Уханова, И.С. Ионова, В.В. Епифанов // Радиотехника. 2017. № 6. С. 49–58.
20. IBM RUP. URL: https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf (дата обращения: 18.09.2020).
21. CPN Tools. URL: <http://cpntools.org/2018/01/16/documentation-2/> (дата обращения: 18.09.2020).

REFERENCES

1. Sapozhkova T.E. Sravnitelnyi analiz podkhodov k modelirovaniu biznes-protsessov [A Comparative Analysis of Approaches to Business Processes Modeling]. *Prikladnaia informatika* [Applied Informatics], 2012, no. 1 (37), pp 14–19.
2. Suiunova G.B., Gaivoronskaia N.A., Polovinko E.V. Metody optimizatsii ekonomicheskikh i sotsialnykh sistem s ispolzovaniem tekhnologii modelirovaniia biznes-protsessov [Methods of Optimizing Economic and Social Systems Using Technologies of Modeling Business Processes]. *Modelirovanie, optimizatsiia i informatsionnye tekhnologii* [Modeling, Optimization and Information Technology], 2018, vol. 6, no. 1, pp. 176–184.
3. Makurkov A.S., Shibano S.V. Notatsii dlia konstruirovaniia aktivnykh pravil [Notes for Active Rules Design]. *Molodezh i nauka: aktualnye problemy fundamentalnykh i prikladnykh issledovani* [Youth and Science. Burning Problems of Basic and Applied Researches]. 2019, pp. 367–370.
4. Virolainen A.V. Modelirovanie biznes-protsessov [Business Process Modeling]. *Luchshaia nauchnaia statia 2019. Sb. st. 24 Mezhdunar. nauch.-issled. konkursa* [Proc. of the 24th Int. Sci. and Research Comp. The Best Research Paper in 2019]. Penza, MTsNS 'Nauka i Prosveshchenie' Publ., 2019, pp. 82–86.
5. Chernyshov A.S. Analiz metodov modelirovaniia avtomatizirovannykh sistem [Analysis of Modeling Computer-Aided Systems]. *Informatsionnye tekhnologii i avtomatizatsiia upravleniia. Sb. tr. konf.* (Omsk, 15–16 maia 2019 g.) [Proc. of Conf. on Information Technologies and Management Automation]. Omsk, 2019, pp. 363–370.

6. Kopp A.M., Orlovskii D.L. Podkhod k analizu i optimizatsii modelei biznes-protsessov v notatsii BPMN [Design Automation of Easy-Tested Digital Finite State Machines]. *Radioelektronika, informatika, upravleniia* [Radio Electronics, Computer Science, Control], 2018, no. 2 (45), pp. 108–116.
7. Sergievskii M.V., Kirpichnikova K.K. Validatsiia i optimizatsiia diagramm klassov UML [Validating and Optimizing UML Class Diagrams]. *Cloud of Science*, 2018, vol. 5, no. 2, pp. 367–378.
8. Bogdan I., Zadorozhnii A. Klassifikatsiia oshibok na UML-diagrammakh, vznikaiushchikh v khode razrabotki IT-proektov [The Classification of Errors on UML-Diagrams Occurring in the Development of IT-Projects]. *Tekhnichni nauki ta tekhnologii* [Technical Sciences and Technologies], 2018, no. 1, pp. 68–78.
9. Kopp A.M., Orlovskii D.L. Analiz ta optimizatsiia modelei biznes-protsesiv u notatsiiakh epc ta bpmn [Analysis and Optimization of EPC and BPMN Business Process Models] *Tekhnichni nauki ta tekhnologii* [Technical Sciences and Technologies], 2018, no. 4 (14), pp. 145–152.
10. Anseeuw J. et al. Design Time Validation for the Correct Execution of BPMN Collaborations. *CLOSER* (1), 2016, pp. 49–58.
11. Ramos-Merino M. et al. A Pattern Based Method for Simplifying a BPMN Process Model. *Applied Sciences*, 2019, vol. 9, no. 11, p. 2322.
12. Claes J., Vandecaveye G. The Impact of Confusion on Syntax errors in Simple Sequence Flow Models in BPMN. *International Conference on Advanced Information Systems Engineering*. Springer, Cham, 2019, pp. 5–16.
13. Lima L., Tavares A., Nogueira S.C. A Framework for Verifying Deadlock and Nondeterminism in UML Activity Diagrams Based on CSP. *Science of Computer Programming*, 2020, p. 102497.
14. Huang H., Peng R., Feng Z. Efficient and Exact Query of Large Process Model Repositories in Cloud Workflow Systems. *IEEE Transactions on Services Computing*, 2015, vol. 11, no. 5, pp. 821–832.
15. Van der Aalst W.M. Everything You Always Wanted to Know about Petri Nets, but Were Afraid to Ask. *International Conference on Business Process Management*. Springer, Cham, 2019, pp. 3–9.
16. Barash M., Okhotin A. Generalized LR Parsing Algorithm for Grammars with One-sided Contexts. *Theory of Computing Systems*, 2017, vol. 61, no. 2, pp. 581–605.
17. Bakinova E. et al. Formal Languages over GF (2). *International Conference on Language and Automata Theory and Applications*. Springer, Cham, 2018, pp. 68–79.
18. Gorokhov A., Grigorev S. Extended Context-Free Grammars Parsing with Generalized LL. *International Conference on Tools and Methods for Program Analysis*. Springer, Cham, 2017, pp. 24–37.
19. Afanasev A.N., Voyt N.N., Ukhanova M.E., Ionova I.S., Epifanov V.V. Analiz konstruktorsko-tehnologicheskikh potokov rabot v usloviakh krupnogo radiotekhnicheskogo predpriiatiia [Design Engineering Workflow Analysis at Large Radio Engineering Enterprises]. *Radiotekhnika* [Radioengineering], 2017, no. 6, pp. 49–58.
20. IBM RUP. Available at: https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf (accessed 18.09.2020).
21. CPN Tools. Available at: <http://cpntools.org/2018/01/16/documentation-2/> (accessed 18.09.2020).