

СИСТЕМЫ АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ

УДК 355.01: 004.056

П.И. Соснин, Ю.А. Лапшов, В.А. Маклаев

ПСЕВДО-КОДОВОЕ ПРОГРАММНОЕ УПРАВЛЕНИЕ ПОТОКАМИ РАБОТ В ПРОЕКТИРОВАНИИ АВТОМАТИЗИРОВАННЫХ СИСТЕМ

Соснин Петр Иванович, заслуженный работник высшей школы РФ, доктор технических наук, профессор, окончил радиотехнический факультет Ульяновского политехнического института. Заведующий кафедрой «Вычислительная техника» УлГТУ. Имеет многочисленные труды в области концептуального проектирования автоматизированных систем. [e-mail: sosnin@ulstu.ru].

Лапшов Юрий Александрович, магистр техники и технологий, окончил Ульяновский государственный технический университет. Аспирант кафедры «Вычислительная техника» УлГТУ. Имеет статьи в области САПР. [e-mail: y.lapshov@mail.ru].

Маклаев Владимир Анатольевич, кандидат технических наук, окончил радиотехнический факультет Ульяновского политехнического института. Генеральный директор ФНПЦ ОАО «НПО «Марс». Имеет статьи в области САПР. [e-mail: mars@mv.ru].

Аннотация

Представляются средства и язык псевдо-кодowego программирования потоков работ (ПР), предназначенные для алгоритмического представления бизнес-процессов (БП), исполняемых в проектировании автоматизированных систем (АС). К специфике языка относится его ориентация на интеллектуальный процессор (роль проектировщика), разделяющий общую работу с компьютерным процессором и с другими исполнителями бизнес-процессов.

Ключевые слова: автоматизированное проектирование, потоки работ, программное управление, псевдо-кодowego программирование.

Petr Ivanovich Sosnin, honored worker of the Higher School of the Russian Federation; Doctor of Engineering, Professor; graduated from the Faculty of Radioengineering of Ulyanovsk Polytechnic Institute; head of the Chair 'Computer Engineering' at Ulyanovsk State Technical University; author of numerous papers in the field of conceptual design of computer-aided systems. e-mail: sosnin@ulstu.ru.

Yury Alexanderovich Lapshov has a Master's degree in Engineering and Technologies; graduated from Ulyanovsk State Technical University; post-graduate student at the Chair 'Computer Engineering' of Ulyanovsk State Technical University; author of articles in the field of CAD. e-mail: y.lapshov@mail.ru.

Vladimir Anatolyevich Maklaev, Candidate of Engineering; graduated from the Faculty of Radioengineering of Ulyanovsk Polytechnic Institute; Director General of Federal Research-and-Production Center Open Joint-Stock Company 'Research-and-Production Association 'Mars'; author of articles in the field of CAD. e-mail: mars@mv.ru.

Abstract

The article presents facilities and a Language for pseudocode programming of workflows, designed for algorithmic representation of business processes to be executed in design of computer-aided systems. The specific character of the Language consists

in its orientation to intelligent processor (designer role) which shares common work with computer processor and other business-process executers.

Key words: computer-aided design, workflow, software, pseudocode programming.

ВВЕДЕНИЕ

Потоки работ используются как базовые модельные представления динамики бизнес-процессов практически во всех современных технологиях разработки автоматизированных систем. На настоящий момент времени опыт моделирования БП с использованием потоков работ представляют разнообразные средства их автоматизации, включающие специализированную графику (например, IDEF, BPMN, eEPC, UML), формальные языки (BPEL, YAWL, XLANG) и инструментальные среды (FloBuilder, COSA, Workflow Factory).

Каждый поток работ должен адекватно представлять совокупность (нормативных) задач проектирования, процесс согласованного решения которых приводит к осуществлению соответствующего БП. Более того, в общем случае в процесс решения может быть вовлечена группа проектировщиков $\{D_j\}$, каждый из которых несёт ответственность за решение назначенных ему проектных задач $\{Z_{ij}\}$ в конкретных условиях. Обобщённое представление потока работ, раскрывающее его структурное содержание, приведено на рисунке 1.

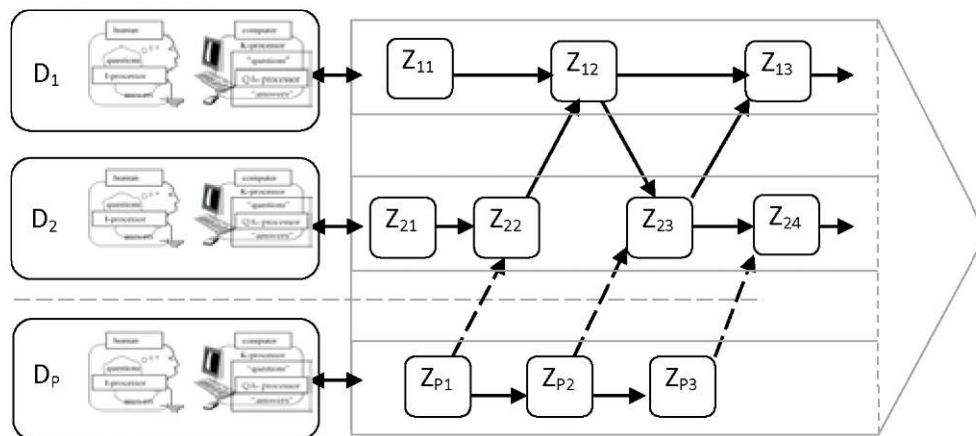


Рис. 1. Пример структуры потока работ

Основное предназначение потоков работ как моделей - автоматизация бизнес-процессов, в которой выделяются автоматизация потоков работ, автоматизация согласованного исполнения задач в потоках и автоматизация решения задач по образцам, причём во всех видах автоматизации компьютеризируется работа лиц, которые вовлечены в исполнение бизнес-процессов [12].

В теории и практике потоков работ, которые не привязаны к конкретным видам деятельности, среди выделенных направлений автоматизации основное внимание уделяется первым двум из-за намеренного абстрагирования от того, в какие действия человека и с чем выливается решение назначенных ему задач. В каждом виде деятель-

ности используемые действия имеют свою специфику, из-за чего к автоматизации решений нормативных задач не применимы типовые средства автоматизации потоков работ.

В проектировании АС, которое осуществляется в человеко-компьютерных средах, человек, следуя инструкциям, решает нормативные задачи, осуществляя шаг за шагом действия с данными и программными конструктами. Автоматизацию решений таких задач логично нацелить на автоматизацию исполнения инструкций, предписанные действия которых исполняются либо человеком, либо компьютером, либо человеком и компьютером совместно.

В статье представлен подход к компьютеризации БП по схемам ПР, в основу которого положено псевдо-кодовое программирование, позволяющее в единообразной форме осуществить автоматизацию по всем отмеченным направлениям. Материализация подхода осуществлена в вопросно-ответной моделирующей среде WIQA (Working In Questions and Answers) [2].

1 АЛГОРИТМИЧЕСКАЯ СПЕЦИФИКА ПОТОКОВ РАБОТ

Как уже отмечалось выше, основное предназначение потоков работ как моделей - автоматизация бизнес-процессов за счёт их компьютеризации. В этом плане активность исполнителей потоков работ разумно связать с согласованной работой совокупности интеллектуальных процессоров, взаимодействующих с используемыми ими компьютерными процессорами.

Для такой интерпретации участия исполнителей в автоматизированных БП достаточно оснований:

1. Каждая задача любого ПР решается (в соответствии с определённым регламентом) исполнителем, использующим в процессе решения свой персональный опыт как в

процессе решения задачи, так и во взаимодействии с исполнителями других задач.

2. Типичным представлением нормативного решения каждой задачи ПР является инструкция (методика), пошаговое исполнение которой имеет алгоритмическое содержание.

3. Для ПР разработана и активно используется на практике библиотека паттернов (образцов), для каждого из которых разработана алгоритмическая схема [3].

Предложенная интерпретация особенно полезна для ПР, осуществляемых в человеко-компьютерных средах, когда продукты деятельности (артефакты) материализуются в форме информационных и алгоритмических конструкторов. Богатейшим примером таких конструкторов являются артефакты инструментальной среды Rational Unified Process (RUP), обеспечивающей разработку автоматизированных систем.

Инструментарий RUP предназначен для коллективной разработки SIS (в российской нормативной базе автоматизированных систем) с помощью системы ПР, включающей 6 основных потоков работ и 3 потока работ поддержки. Для построения артефактов в этой системе разработчикам приходится создавать и объединять в соответствующий артефакт определённые текстовые, табличные и графические конструкторы.

Таким образом, при использовании ПР в человеко-компьютерной среде к уже отмеченной алгоритмической специфике таких моделей добавляется оперативная работа исполнителей только с текстовыми, табличными и графическими объектами, которые получают материализованное представление в компьютерной памяти и визуально доступны на мониторах компьютеров.

2 СПЕЦИФИКА ПОТОКОВ РАБОТ В ПРОЕКТИРОВАНИИ АВТОМАТИЗИРОВАННЫХ СИСТЕМ

В любых его приложениях проектирование АС является деятельностью команды проектировщиков $K(\{D_r\})$, которая, автоматизируя определённую систему бизнес-процессов $E(\{BP_r\})$, должна **согласованно решать** определённую **совокупность задач** $\{Z_j\}$ используемой **технологии** T^* . По сути дела, в такой деятельности систему решений задач $S(\{Z_{mj}\}, t)$, вложенных в $\wedge(\{BP_r\})$, пытаются представить через их «разложение» в базе задач технологии T^* .

В любой технологии T^* её задачи объединяют в потоки работ $\{W^m(\{Z_r\})\}$, а потоки работ объединяют в подсистемы $G^r(\{W^m(\{Z_r\})\})$ в соответствии со специализацией этапов проектирования. Задачи представлены инструкциями, раскрывающими **схемы действий**, каждая из которых должна быть адаптирована к соответствующим бизнес-процессам АС.

Необходимость такой адаптации задач $\{Z_{mj}\}$ вводит в процесс проектирования специфический класс творческих **задач адаптации** $\{Z^A\}$, включающих задачи когнитивного анализа, оценивания, принятия решений и другие задачи, требующие от проектировщика ситуативной (незапланированной заранее) интеллектуальной активности

[4]. Более того, такая адаптация должна осуществляться проектировщиками согласованно. Именно необходимость **согласованной интеллектуальной адаптации задач** в реализации потоков работ технологии определяет важнейшую их специфику.

Реальная практика такой адаптации является источником многочисленных и разнородных ошибок и «дефектов», оказывающих негативное воздействие на процесс проектирования и его результаты. Основным источником негативов является человеческий фактор [5], а вернее - **неудовлетворительное исполнение проектировщиками их функций в технологических потоках работ**, обусловленное **недостаточной степенью автоматизации активности проектировщиков** в решении задач адаптации. Такое положение дел, в первую очередь, указывает на **несовершенство** существующих технологий $\{T\}$, индикатором чего является мировая статистика чрезвычайно низкой степени успешности разработок АС (около 35 %) [6].

За последние 20 лет использовались различные средства совершенствования технологий класса $\{T\}$, например, модели совершенствования профессиональной деятельности [7], унифицированный язык моделирования [1], базы опыта [5] и фабрики опыта [11, 13], нормативные средства описания и реализации архитектурных решений [8]. Шаг за шагом появлялись новые полезные средства и совершенствовались уже освоенные, однако степень успешности разработок АС почти не повышалась. А значит, проблема успешности проектирования АС всё ещё существует и поиски путей её решения за счёт технологического совершенствования деятельности проектировщиков актуальны.

3 СОБЫТИЙНОЕ СВЯЗЫВАНИЕ ПОТОКОВ РАБОТ В ПРОЕКТИРОВАНИИ АВТОМАТИЗИРОВАННЫХ СИСТЕМ

Для выделения одного из направлений такого совершенствования проведём анализ отношений между нормативными задачами $\{Z_r\}$ потока проектных работ $W^n(\{Z_r\})$ и задачами их адаптации к предметной области разрабатываемой АС. В общем случае для решения каждой нормативной задачи Z_r потока $W^n(\{Z_r\})$ в его конкретном \wedge -исполнении $W^{nk}(\{Z_{rj}\})$ может потребоваться решить группу задач адаптации, каждую из которых Z^A логично квалифицировать как подзадачу, подчинённую задаче $Z_{цк}$.

А значит, процесс решения задачи Z_{njk} в потоке $W^{nk}(\{Z_{rj}\})$ можно представить как решение задачи Z_{njc} ($S(\{Z^A_{rj}\}, t)$, t), задачная структура которой ситуативно изменяется за счёт включения в процесс решения задач адаптации $S(\{Z^A_{rj}\}, t)$, понадобившихся проектировщику D^c , ответственному за решение задачи Z_{njc} .

Следует отметить, что решение задачи Z_{njc} должно осуществляться в контексте других задач потока работ W^{nk} . Условия такого контекста представим проекцией $P^n_{jk}(W^{nk}_{rj}, Z_{njc}, t)$ потока W^{nk} , на задачу Z_{njc} , связав с таким конструктором динамическую совокупность событий

$E_{JK}^n(\{e_{JK}^n(t)\})$, которые порождаются исполнением задач $W_{K(\{Z_{jk}\})}^n$ и должны приниматься в расчёт при решении задачи Z_{JK} . Для всех задач потока $W_{K(\{Z_n\})}^n$ совокупности событий вида $E_{JK}^n(\{e_{JK}^n(t)\})$ порождаются для каждой из задач потока и образуют единое множество $\{P_{.k}^n(W_{.k}^n, z_{nj}, t)\} = \{J e_{jk}^n(t)\}$.

Кроме того, решение задачи Z должно осуществляться в контексте автоматизируемой системы бизнес-процессов $S(\{BIM_m\})$, а вернее в контексте той задачи или задач системы $S(\{Z_{mi}\}, t)$, для которой (-ых) задача Z включена в процесс проектирования АС. В общем случае решения задач совокупности $\{Z_{mi}\}$ либо модернизируются, либо строятся в процессе проектирования с ориентацией на их автоматизацию. Другими словами, задачи совокупности $\{Z_{mi}\}$ адаптируются под их автоматизированное исполнение на этапе эксплуатации АС. А значит их контекст для задачи Z тоже можно представить в событийной форме $P_{nj,k}^m(\{Z_{mi}\}, Z_{nj,k}, t) = E_{nj,k}^m(\{e_{nj,k}^m(t)\})$.

Более того, потоки работ исполняемой технологии $T^*(\{G^r(\{W_{k(\{Z_{nj,k}\})}\})\}, t)$ образуют связанное целое, связность которого также логично специфицировать с помощью проекций $T^*(\{G^r(\{W_{k(\{Z_{nj,k}\})}\})\}, t)$, представленных соответствующими системами событий $E_{nj,k}^m(\{e_{nj,k}^m(t)\})$, которые должны приниматься в расчёт в процессе проектирования АС.

Введение совокупностей событий, связывающих задачи потоков $\{e_j(t_s)\}$, задачи потоков работ $\{e_{j,k}^m(tV)\}$ разрабатываемой SIS, потоки этапов и этапы в используемой технологии $\{e_{nj,kp}^r(t_w)\}$ открывает возможность для динамической систематизации задач

$$T^*(t) = S(\{J e_{n,kp}^m(t)\}), \quad \{E^m J_x e_{jkp}^n(t)\},$$

в текущем состоянии (в текущий момент времени t) процесса проектирования SIS.

4 ПОРОЖДЕНИЕ И ИСПОЛЬЗОВАНИЕ СОБЫТИЙ

Для введения и использования динамической систематизации следует определиться с порождением событий и их связыванием в конструкты типа E_{jk}^n , $E_{nj,k}^m$ и $E_{nj,k}^r$. Основным источником событий являются проектировщики, решающие назначенные им задачи. В этом плане (с позиции назначения ответственных) в коллективе проектировщиков должны быть назначены ответственные за реализацию каждого потока работ, групп потоков работ, этапов и проекта АС в целом, что позволяет интерпретировать названные конструкты как «задачи», но соответствующих типов ZW и ZG . Такая интерпретация позволяет ввести определённую унификацию в представление задачи и поведение проектировщиков, ответственных за задачи проекта АС. В частности, для задач можно будет ввести и использовать единообразную индексацию ZJ , в которой символ « Z » указывает на объект типа «задача», а символ « J » является (автоматически вычисляемым и приписываемым) уникальным составным индексом, регистри-

рующим «место» задачи среди задач проекта.

Необходимость адаптации задач всех названных типов требует предоставить каждому члену коллектива проектировщиков, решающему любую из назначенных ему задач, свободу в выявлении необходимой ему совокупности задач адаптации $\{Z^i\}$, формулировке их постановок и решения. Разумеется, проектировщикам необходимо предоставить свободу в рамках обязательств, обусловленных коллективной деятельностью и нормативами используемой технологии.

Представим специфику обязательств и их учёта:

1. В общем случае процесса проектирования любому члену D_p коллектива $K(\{D_p\})$ может быть назначена ответственность за решение совокупности задач $C(\{ZJ(D_p)\}, t)$, которая может динамически изменяться. Факты назначения логично учесть в атрибутике задач и в модели коллектива (в представлении его организационной структуры для узла, моделирующего проектировщика D_p в профессиональном плане).

2. Если задача $ZJ(D_p)$ относится к классу задач Z^n , то есть к задачам бизнес-процессов разрабатываемой АС (назовём такие задачи «предметными»), и проектировщику D_p придётся искать её решение (новая задача, реинжиниринг), то проектировщик:

- свободен в порождении её предметных подзадач и оказавшихся необходимыми задач адаптации;

- частично свободен в использовании технологических задач типа $Z_{nj,k}^r$

- связан плановыми обязательствами по решению задачи $ZJ(D_p)$, принятыми в коллективе.

3. Если задача $ZJ(D_j)$ относится к классу задач $n^{j,k}$, то есть к задачам бизнес-процессов технологии, то проектировщик:

- свободен в порождении задач адаптации, оказавшихся необходимыми;

- связан отношениями между задачами, регламентированными в соответствующем бизнес-процессе;

- связан плановыми обязательствами по решению задачи $ZJ(D_p)$, принятыми в коллективе.

4. Если задача $ZJ(D)$ представляет бизнес-процесс этапа технологии T^* или этап, то член коллектива, ответственный за этот процесс:

- свободен в порождении задач адаптации, оказавшихся необходимыми;

- обязан действовать в соответствии с нормативными правилами регламента, которые должны соблюдаться для этого бизнес-процесса или этапа;

- связан плановыми обязательствами по решению задачи $ZJ(D_p)$, принятыми в коллективе.

Таким образом, независимо от того, какую роль исполняет член коллектива проектировщиков, в его активности имеются составляющие, при осуществлении которых он свободен в их планировании и псевдопараллельном исполнении. В управлении такой работой проектировщику поможет «Система прерываний человеко-компьютерной

деятельности», настроенная на подходящую типологию $T(\{B_i\})$ прерываний $\{B_i\}$ псевдопараллельного решения задач адаптации. В рамках данной активности проектировщик будет решать совокупность задач $\{ZK(D_p)\}$, порождая и используя совокупность событий $E^a_k(\{e^k(t)\})$, основными типами которых являются «Задача сформулирована», «Приступить к решению задачи», «Прервать процесс решения» и «Решение задачи завершено». Ещё раз отметим, что рассматриваемый класс событий порождается проектировщиком и используется им для управления собственной активностью в собственных целях.

По глубокому убеждению авторов, оперативную работу с задачами $\{ZK(D_p)\}$, то есть их формулировку, поиск решений, их кодирование и исполнение, следует осуществлять в формах программирования, используя подходящий псевдо-кодовый язык. Более того, псевдо-кодовые программы для задач $\{ZK(DP)\}$ должны быть ориентированы на их исполнение проектировщиками.

Для использования событий классов $E^n_{jR'}$, E^m_{ijk} и E^r_{ijk} в управлении процессами решения задач всех остальных типов следует:

- создать условия для использования «Системы прерываний человеко-компьютерной деятельности» на любом рабочем месте коллектива проектировщиков;
- расширить типологию прерываний $T(\{B_i\})$ за счёт приращений $AT(\{B_j(E^m)\})$, $DT(\{B_k(E^n)\})$ и $DT(\{B_l(E^o)\})$;
- создать совокупность средств порождения событий за счёт правил, регламентирующих отношения между задачами в технологических бизнес-процессах, на этапах и в проекте АС в целом.

Обобщая вышесказанное, отметим, что представленная версия порождения и использования событийной систематизации потоков работ вводит в процессы решения задач и управления процессами дополнительные полезные составляющие, совершенствующие процесс проектирования за счёт автоматизации работ по адаптации

технологии к специфике разрабатываемой АС. Версия доведена до её реализации в инструментальной моделирующей среде WIQA.

5 ПРЕДСТАВЛЕНИЕ ПОТОКОВ РАБОТ В МОДЕЛИРУЮЩЕЙ СРЕДЕ WIQA

Инструментальная среда WIQA изначально разрабатывалась для моделирования совокупностей типовых и предметных проектных задач, которые приходится решать разработчикам АС. Как источник типовых задач были использованы типовые задачи потоков работ концептуального проектирования в технологии Rational Unified Process [2, 9]. По предназначению и содержанию WIQA разрабатывалась как специализированная АС технологического типа.

Одной из важнейших особенностей концептуального проектирования в среде WIQA является потенциальное применение всех средств этого инструментария к любой задаче Z_i дерева $T(\{Z_i\})$ проектных задач $\{Z_i\}$, если в этом будет необходимость. В таком применении задача Z_i представляется её вопросно-ответной моделью (QA-моделью, $QA(Z_i)$), структурирующей процесс решения задачи в форме вопросно-ответных рассуждений. Интерфейсная форма доступа к дереву задач проекта и их QA-моделям представлена обобщённо на рисунке 2.

Информационный потенциал дерева задач $G(\{Z_i\})$ и связанного с ним множества моделей $\{QA(Z_i)\}$ достаточен для представления текущего состояния совокупности потоков работ $\{W_m\}$ любого проекта $\Pi(0(\{Z_i\}), \{QA(Z_i)\}, R(\{D_p\}), t)$, разрабатываемого коллективом $K(\{D_p\})$ проектировщиков $\{D_p\}$. Для представления и использования конструкта $K(\{D\})$ в комплекс WIQA встроено расширение (плагин), названное «Оргструктура».

На основе этого потенциала авторами разработан комплекс средств для представления проектов, разрабатываемых в инструментальной среде WIQA, в виде динамической схемы $\mathcal{F}(\{W_m\}, t)$ потоков работ, предназначенной для управления процессами проектирования. В схеме

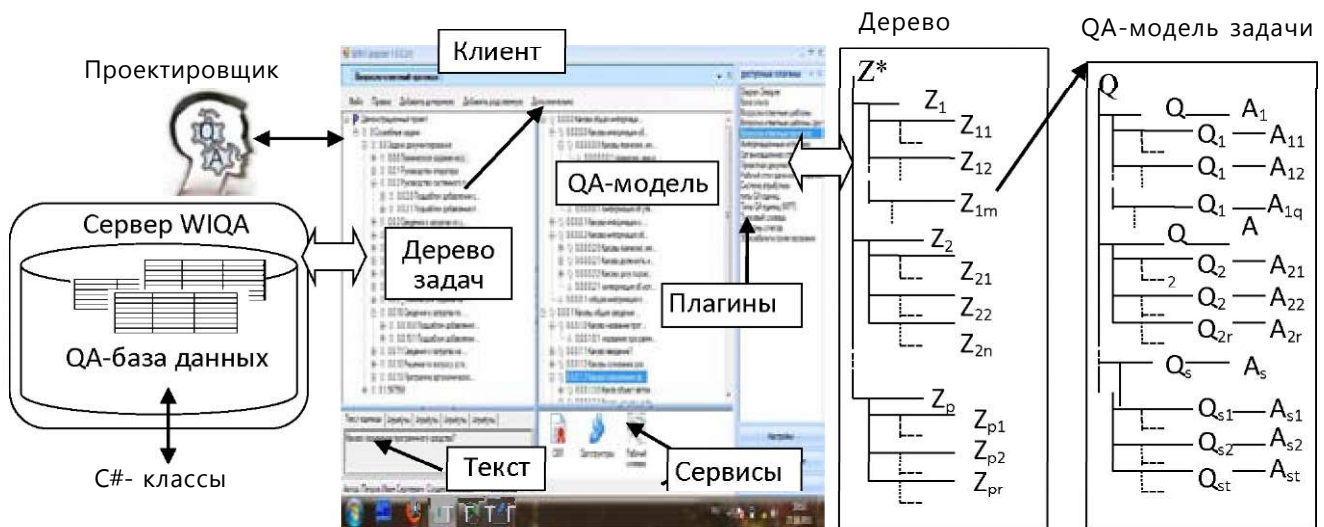


Рис. 2. Структура и представление дерева задач и их QA-моделей

$\mathcal{L}(\{W_m\}, t)$ с каждым потоком W_m связана определённая задача Z_i дерева задач, которой приписан тип «поток работ» и символьное обозначение типа W .

Таким образом, в дерево задач может быть введена разметка, регистрирующая, «какие задачи включены в каждый выделенный поток работ в текущем состоянии проекта». Введение разметки приводит к новой версии $n(G(\{Z_i\}, \{W_m\}), \{QA(Z)\}, \{QA(W)\}, K(\{DJ\}), t)$ представления проектов АС. В разработанном комплексе средств управления проектами эта версия используется как основной источник информации для решения задач управления.

На рисунке 2 отражено, что WIQA реализована как клиент-серверный комплекс инструментов, вопросно-ответная база (QA-база) данных которого размещена на сервере. База объектно-ориентированная, позволяющая (за счёт объектно-реляционных преобразований) строить аналоги хранимых объектов (например, задачи) в виде C#-классов.

6 ПРОГРАММНОЕ УПРАВЛЕНИЕ ПОТОКАМИ РАБОТ

По ходу развития инструментария WIQA средства управления потоками работ совершенствовались, и на настоящий момент времени они включают средства событийной систематизации, представленные выше. Для осуществления такой систематизации и её использования в управляющих целях был разработан ряд плагинов, обеспечивающих:

- псевдо-кодовое программирование задач, ориентированное на исполнение программ проектировщиком, выполняющим роль интеллектуального процессора (I-процессора)

и использующим компьютерный процессор (K-процессор) как подчинённый процессор;

- управление прерываниями человеко-компьютерной деятельности, в частности управление прерываниями при исполнении псевдо-кодовых программ для технологических задач, кодирующих нормативные инструкции;

- оперативное использование паттернов потоков работ, псевдо-кодовые аналоги которых хранятся в специальной библиотеке.

Кроме того, для представления событий всех типов используется атрибутика объектов, входящих в дерево задач и QA-модели, в частности их уникальные индексные имена, имена проектировщиков, время последней модификации и состояния («в работе», «прервана» и «завершена»). Для учёта временных характеристик работ (решений задач всех типов) и их соответствия договорённостям разработана специализированная система «Контроля поручений».

Учитывая тот факт, что в основу событийной систематизации положено запланированное и оперативное (псевдо-кодовое) программирование, её использование в управляющих целях осуществляется в форме программ-

ного управления потоками работ.

7 СПЕЦИФИКА ПСЕВДО-КОВОГО ЦЫКА WIQA

В программном управлении потоками работ задачи потока представлены в форме программ на псевдо-кодовом языке (L^{WIQA}), получившем название WIQA. Это название отражает не только тот факт, что язык встроен в инструментарий WIQA, но и то, что он явно и конструктивно на-следует вопросно-ответную структуризацию человеко-компьютерных взаимодействий.

Язык L^{wiqa} использовался для программирования задач диагностики приборов, каталогизации документов, оценивания и принятия решений. Очередным его приложением является комплекс средств управления потоками работ.

Специфику языка L^{WIQA} определяет «запись» операторов исходного кода программ на носителе, функции которого выполняет определённая вопросно-ответная пара QA-объектов. При использовании носителей такого типа операторами псевдо-кодового языка наследуются свойства QA-объектов и система операций над ними.

Текст оператора регистрируется в текстовом поле соответствующего вопроса Q и наследует возможности его визуализации и оперативного редактирования. Индексное имя способно выполнять роль адреса в QA-памяти среды WIQA. Автоматически регистрируются ав-

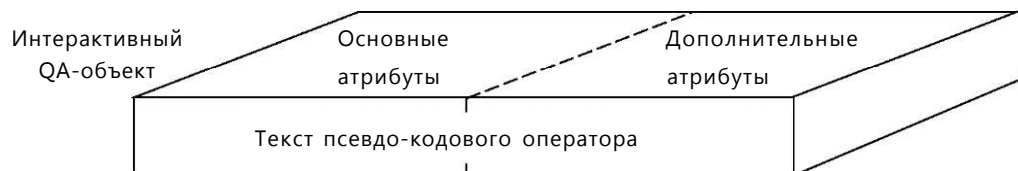


Рис. 3. Материализация псевдо-кодового оператора

тор псевдо-кодового оператора (QA-объекта), каждое изменение и его момент времени (по машинным часам). Другие нормативные атрибуты QA-объектов позволяют зарегистрировать и использовать для любого оператора другую полезную информацию [15].

Более того, в инструментарий WIQA встроены средства объектно-реляционных преобразований QA-объектов, позволяющие приписать любому QA-объекту любое количество дополнительных атрибутов, полезных для реализации решаемых задач. Материализация текста оператора в образной форме представлена на рисунке 3.

Отмеченные средства позволяют любому традиционному оператору псевдо-кодового алгоритмического языка приписать полезные нормативные и дополнительные атрибуты соответствующей пары QA-объектов. Наследование операторами языка свойств материала, на котором они «записываются», существенно изменяет традиционную грамматику псевдо-кодового языка, предоставляя возможность присоединить к ней метаграмматику $L^M(L^{WIQA})$, созданную программистом для программируемой задачи. Такая возможность одна из важнейших причин, по

которой языку было решено присвоить собственное имя *WIQA*. Ещё одной принципиальной особенностью языка L^{WIQA} является его ориентация на создание программ, исполняемых I-процессором совместно с K-процессором.

Во всём остальном язык L^{WIQA} подобен другим языкам псевдо-кодированного программирования. В этом языке поддерживаются объявления традиционных типов данных и использование привычных операторов (Assign, If_Then_Else, While_Do и другие операторы). Язык L^{WIQA} относится к классу объектно-ориентированных языков с элементами программирования баз данных. Следующий фрагмент псевдо-кодированной программы диагностики прибора (без объяснений) демонстрирует элементы грамматики языка L^{WIQA} .

Q 1.22.8 PROCEDURE &Тест_ППВ&

Q 1.22.8.1 *выбрать пункт меню «ППВ»*

Q 1.22.8.2 *выполнить проверку сенсорной панели ВМС касанием пальцем поочередно всех клеток экрана ВМС, цвет клетки должен меняться, а отметка маркера должна отображаться в зоне выбранной клетки экрана ВМС*

Q 1.22.8.3 INPUT &PPVCOLOR&

Цвет клетки изменяется при касании пальцем экрана ВМС?

0 - нет, 1 - да

A 1.22.8.3.1 1

Q 1.22.8.4 IF &PPVCOLOR& == 1 THEN GOTO &LASTPPV&

Q 1.22.8.5 CALL &Отключить_прибор&

Q 1.22.8.6 *снять переднюю крышку приборного шкафа*

Q 1.22.8.7 CALL &Заменить_ВМС&

Q 1.22.8.8 CALL &Включить_прибор&

Q 1.22.8.9 CALL &Установить_режим_выборочно-го_теста&

Q 1.22.8.10 RETURN

Q 1.22.8.11 LABEL &LASTPPV&

Q 1.22.8.12 *Нажать комбинацию клавиш «CTRL-P»*

Q 1.22.8.13 ENDPROC &Тест_ППВ&

8 СРЕДА УПРАВЛЕНИЯ ПОТОКАМИ РАБОТ

Язык WIQA, как и любой алгоритмический язык, неотделим от средств, обеспечивающих его употребление в намеченных целях. Для языка WIQA в число таких основных целей включены:

1. Рациональное управление выполнением потоков работ в человеко-компьютерных средах.
2. Псевдо-кодированное программирование задач потоков, ориентированное на их согласованное решение интеллектуальным и компьютерным процессорами.

Для согласованного достижения отмеченных целей в системе WIQA используется комплекс инструментальных средств.

В состав комплекса входят: интерпретатор псевдо-кодированных программ, обслуживающий активность интеллектуального процессора (пошаговое исполнение методик бизнес-процесса их исполнителем); компилятор псевдо-кодированных программ, обеспечивающий их автоматическое исполнение процессором компьютера; система управления потоками работ, представлена на рисунке 4.

В системе управления потоками используются компоненты: «Оргструктура» (назначение задач команде проектировщиков), «Управление поручениями» (приписывание временных характеристик исполняемым работам), «Управление прерываниями» (параллельное и псевдопараллельное решение задач в потоках) и «Канбан» [16], а также инструментальный диаграмм Ганта.

ЗАКЛЮЧЕНИЕ

Разработанные и представленные в статье средства управления потоками работ предоставляют возможность единообразного подхода к автоматизации согласованного управления потоками работ, задачами в потоках и действиями проектировщиков в решении нормативных задач на базе псевдо-кодированного программирования в

инструментальной среде WIQA. Такая возможность обеспечивается событийной систематизацией потоков работ и спецификой псевдо-кодированного языка, ориентированного на исполнение программ проектировщиками.

Событийная систематизация введена для управления процессами адаптации потоков работ технологии к потокам работ разрабатываемой АС. Спецификации языка и его выразительные возможности нацелены на доступ к опыту проектировщиков и доступ к моделям опыта. Для структуризации до-

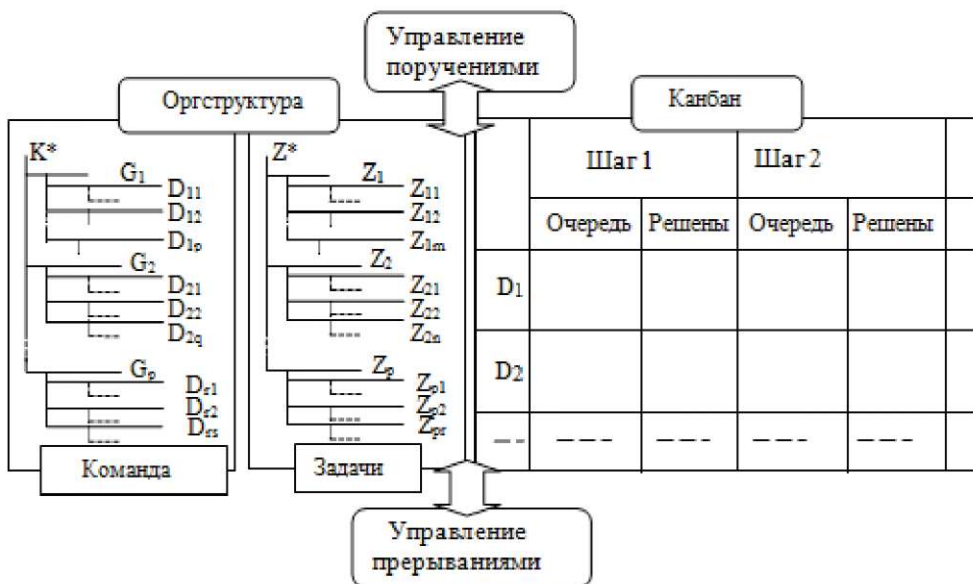


Рис. 4. Система управления потоками работ

ступного опыта используются модели прецедентов.

В оперативном программировании задач адаптации проектировщик выходит за рамки технологического регламента и самостоятельно организует их псевдопараллельное решение с использованием системы «Управления прерываниями». Для организации коллективной работы кроме системы «Управления прерываниями» используется система «Управления поручениями» и система средств «Канбан».

Разработанный комплекс средств псевдо-кодowego программного управления потоками работ вводит в процессы решения задач и управление процессами дополнительные полезные составляющие, совершенствующие процесс проектирования за счёт автоматизации работ по адаптации технологии к специфике разрабатываемой АС.

СПИСОК ЛИТЕРАТУРЫ

1. Кролл, П. Rational Unified Process - это легко: Руководство по RUP для практиков / П. Кролл, Ф. Крачтен. - М.: КУДИЦ-Образ, 2004. - 427 с.
2. Соснин, П.И. Вопросно-ответное моделирование в разработке автоматизированных систем /П.И. Соснин. - Ульяновск: УлГТУ, 2007. - 333 с.
3. Aalst W.M.P., Hofstede A.H.M. and M. Dumas. Patterns of Process Modeling./ In Process-Aware Information Systems: Bridging People and Software through Process Technology, Wiley & Sons, 2005, pp. 179-203.
4. Alexopoulou N., Nikolaidou M., Anagnostopoulou D. and Martakos D. An event-driven modeling approach for dynamic human-intensive business processes, In Proc. of Business Process Management Workshop, (2009), 393-404.
5. Basili A.V., Lindvall M. and Costa P. Implementing the experience factory concepts as a set of experience bases, In Proc. of the 13 th International Conference on Software Engineering & Knowledge Engineering, (2001), 102-109.
6. Charette R.N. Why software falls, IEEE Spectrum, 42(9), (2005), 36-43.
7. Chrissis M.B., Konrad M. and Shrum S. CMMI for Development: Guidelines for Process Integration and Product Improvement, SEI Series in Software Engineering, 2007.
8. IEEE. IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. Institute of Electrical and Electronics Engineers, Sept. 2000. IEEE Std 1471-2000.
9. Kroll P., Kruchten Ph. The Rational Unified Process Made Easy: A Practitioners Guide to the RUP. Addison-Wesley, 2003.
10. Held M. and Blochinger W. Structured collaborative workflow design, Future Generation Computer Systems, 25(6), (2009), 638-653.
11. Henninger S. Tool Support for Experience-based Software Development Methodologies, Advances in Computers, 59, (2003), 29-82.
12. Oren E. and Haller A. Formal frameworks for workflow modeling/ DERI Technical Report 2005-04-07, April 2005.
13. Ras E., Rech J. and Weber S. Knowledge services for experience factories, In Proc. of the 5th Conference on Professional Knowledge Management, (2009), 232-241.
14. Sosnin P. Question-Answer Shell for Personal Expert System // Chapter in the book "Expert Systems for Human, Materials and Automation." Published by Intech, (2011), pp. 51-74.
15. Sosnin P. Question-Answer Approach to Human-Computer Interaction in Collaborative Designing. Chapter in the book "Cognitively Informed Intelligent Interfaces: Systems Design and Development" Published IGI Global, (2012), pp. 157-176.
16. Wang J. X . Kanban: Align Manufacturing Flow with Demand Pull, Chapter in the book: Lean Manufacturing Business Bottom-Line Based, CRC Press (2010), 185-204.