

Учреждение образования  
“БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ”

УДК 628.52.012.011.56

Юсеф Ибрахим Юсеф ДАРАДКЕХ

ИНТЕРПРЕТАЦИЯ ПОЛИМОРФНЫХ СЕТЕВЫХ МОДЕЛЕЙ  
В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ РЕАЛЬНОГО ВРЕМЕНИ

05.13.11 – Математическое и программное обеспечение  
вычислительных машин, комплексов и компьютерных сетей

Автореферат диссертации  
на соискание ученой степени кандидата технических наук

Минск 2005

Работа выполнена на кафедре информационных технологий автоматизированных систем Учреждения образования “Белорусский государственный университет информатики и радиоэлектроники” (БГУИР).

Научный руководитель: к.т.н., доцент Ревотюк М.П.,  
кафедра информационных технологий  
автоматизированных систем, БГУИР

Официальные оппоненты: д.т.н., профессор Суходольский А.М.,  
лаборатория “Информационные технологии  
и системы” научно-исследовательской части  
БГУИР

к.т.н., с.н.с. Григянец Р.Б.,  
Государственное Научное Учреждение  
“Национальный центр информационных  
ресурсов и технологий” Национальной  
Академии Наук Беларуси

Оппонирующая организация: Учреждение образования “Белорусский  
государственный университет”

Защита состоится 19 января 2006 г. в 14 часов на заседании совета по защите диссертаций Д 02.15.04 при Учреждении образования “Белорусский государственный университет информатики и радиоэлектроники” по адресу 220013, Минск, ул. П. Бровки, 6, БГУИР, 1 уч. корпус, тел. 239-89-89.

С диссертацией можно ознакомиться в библиотеке Белорусского государственного университета информатики и радиоэлектроники.

Автореферат разослан 16 декабря 2005 года.

## ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

**Актуальность темы диссертации.** Современные технологии эволюционного конкретизирующего программирования (объектно-ориентированное и шаблонно-ориентированное), опережая по гибкости методы построения моделей систем, предоставляют механизмы (наследование, полиморфизм, пакетирование, шаблоны), вполне обеспечивающие возможность интерпретации различных формальных подходов в терминах языковых конструкций. Не порождая рецептов конструирования, такие технологии существенно определяют интерфейсы созданных и используемых в качестве стандартных компонент создаваемых систем.

Объективная сложность управления производственными процессами, преодолеваемая методами структурно-функциональной декомпозиции, порождает на организационном уровне задачи управления взаимодействующими асинхронными дискретными процессами. Наиболее адекватными моделями таких процессов являются имитационные модели. При этом вопрос выбора вида модели обычно не ставится. Ставший стандартным в последнее время UML (Unified Modeling Language) решает задачу спецификации программной системы в терминах сетей Петри либо конечных автоматов. Однако вопросы поиска законов управления оказываются вне модели.

Вместе с тем, анализ опыта применения алгоритмически интерпретируемых сетей Петри и их расширений показывает, что формально определяемые переходы в таких сетях можно связать процессами поиска решений, дополняя их слоями конкретизации описания на основе полиморфизма функций переходов. Решение подобной задачи предполагает разработку и представление инвариантных элементов сетевого описания параметризованными иерархиями полиморфных классов.

Шаблоны классов и функций могут рассматриваться как средство представления системы, аккумулирующее удачные приемы и методы представления пространства ее состояния, а также поиска решений. Возможность специализации шаблонов позволяет организовать эволюционный процесс уточнения схем вычислительных процессов, начиная от общих процедур логического вывода или исчерпывающего поиска, включая возможности автоматического распараллеливания.

Таким образом, тема исследования методов создания единой формально согласованной и отображаемой на исполнительный уровень вычислительных сред модели, позволяющих достичь практической реализуемости сквозных технологий построения открытых систем на основе принципов конкретизации описания, является актуальной.

**Связь работы с крупными научными программами, темами.** Исследования проводились в рамках научно-исследовательской госбюджетной темы № 01-2016 «Исследовать проблемы системного анализа, обработки информации и управления на основе перспективных компьютерных

технологий”(номер госрегистрации 2004306) кафедры информационных технологий автоматизированных систем БГУИР, а также хоздоговорной НИР № 02-1113 “Разработка методов и средств защиты прикладного программного обеспечения от несанкционированного использования”(номер госрегистрации 20023923), выполненной НИГ 3.1 “Автоматизированные системы управления” НИЧ БГУИР.

**Цель и задачи исследования.** Целью данной работы является разработка методов и средств эффективной реализации программного обеспечения систем управления дискретными производственными процессами, включающими фазы планирования в ускоренном и исполнения в реальном режиме времени.

Для достижения указанной цели необходимо решить следующие задачи:

- 1) разработать шаблоны классов объектно-ориентированной спецификации процессов управления взаимодействующими процессами на организационно-технологическом уровне;
- 2) разработать шаблоны процедур оптимизации управления на алгоритмически интерпретируемых полиморфных сетях;
- 3) исследовать методы объектно-ориентированной спецификации кооперативных схем решения комбинаторных задач выбора на распределенных вычислительных средах;
- 4) рассмотреть вопросы обеспечения контроля условий целостности и безопасности агентов распределенных вычислений на локальных сетях.

**Объект и предмет исследования.** Объектом исследований настоящей работы является программное обеспечение систем управления объектами с дискретным характером поведения и технология его разработки. Предмет исследования – методы формализации и реализации процессов поиска оптимальных решений и координации в открытых на уровне цели управления взаимодействующих системах на основе иерархии шаблонов полиморфных классов.

**Гипотеза.** Сочетание классических методов имитационного моделирования и оптимизации дискретных процессов, объектно-ориентированного проектирования и технологий конкретизирующего программирования позволит получить новые научно-технические результаты и повысить уровень проектно-конструкторских решений.

**Методология и методы проведенного исследования.** Решение рассматриваемых в диссертации задач базируется на методологии объектно-ориентированного моделирования и программирования, методах имитационного моделирования, теории сетей Петри и теории графов.

**Научная новизна и значимость полученных результатов.**

1. Обоснована схема шаблонов классов интерпретаторов сетевых описаний общего вида, отличающихся от известных моделей дискретных процессов новой реализацией принципа детализации описания, основанного на использовании компактных рекуррентных схем формализации.
2. Предложенные процедуры группового поиска кратчайших путей на графах, учитывающие ограничения на структуры путей и predeterminedные

решения, оптимизированные по критерию “память-быстродействие”, позволяют снизить степень вычислительной сложности от 2 до 1.25, что существенно для реализации быстродействующих систем типа “клиент-сервер”.

3. Реализованные методы разделения и синхронизации потоков данных определения подмножеств вариантов при решении задач оптимизации решений по схеме кооперации ресурсов вычислительной среды позволяют улучшить динамические характеристики процесса поиска решений в реальном времени.

4. Предложены и обоснованы схемы шаблонов классов многоуровневой автоматически устанавливаемой защиты исполняемых модулей, позволяющих обеспечить в реальном времени контроль условий целостности и безопасности агентов распределенных вычислений.

#### ***Практическая значимость полученных результатов.***

1. Предложенные полиморфные расширения сетей Петри и переходных систем позволяют интегрировать известные формальные схемы на уровне программной реализации в рамках технологически совместимой с современными системами программирования модели *PME (Property, Method, Event)*, а интерпретатор сетевых описаний общего вида является шаблоном проектирования многоуровневых систем управления.

2. Построены эффективные в смысле “память-быстродействие” алгоритмы поиска и представления кратчайших путей на абстрактных сетях, допускающие учет ограничений. Разработанные на базе полученных алгоритмов комплексы программных средств прошли апробацию на реальных данных модернизируемой системы управления маршрутизацией потоков на Новополоцком нефтеперерабатывающем заводе “Нафтан”.

3. Обоснованы и реализованы объектно-ориентированные схемы многоуровневой статической и динамической защиты разрабатываемых программ, обеспечивающие автоматизацию установки средств контроля условий целостности и безопасности.

4. Результаты диссертационной работы внедрены и использованы в НПРУП “КВАНТ-АС” и ИАЦ РУП НВЦ “БЕЛЭКСПО”, а также учебном процессе на кафедре ИТАС БГУИР.

#### ***Основные положения диссертации, выносимые на защиту.***

1. Уточненные модели представления иерархических процессов императивного управления в терминах объектно-ориентированных полиморфных расширений сетей Петри, неоднородных переходных систем общего вида и эффективные процедуры их алгоритмической интерпретации.

2. Модели и быстродействующие алгоритмы поиска кратчайших путей на графах с ограничениями.

3. Шаблоны классов реализации кооперативных схем решения задач выбора оптимальных вариантов на распределенных вычислительных средах с контролем условий целостности и безопасности.

**Личный вклад соискателя.** В настоящую диссертационную работу вошли результаты как личных исследований автора (развитие аппарата описания иерархических процессов на основе полиморфных расширенных сетей Петри и их алгоритмическая интерпретация в рамках объектно-ориентированного подхода, экспериментальные исследования схем кооперации ЭВМ для решения задач поиска оптимального управления), так и его совместной деятельности с научным руководителем к.т.н., доцентом Ревотюком М.П. (концептуальные схемы объектно-ориентированной модели процесса императивного управления и безопасности распределенных вычислений, аппарат описания расширенных сетей Петри), а также соавторами научных трудов (макетирование шаблонов классов процедур моделирования и оптимизации и экспериментальная проверка их эффективности).

**Апробация результатов диссертации.** Основные результаты работы докладывались на I и II Международных конференциях “Информационные системы и технологии” (Минск, IST’2002; Минск, IST’2004), региональной конференции молодых ученых и студентов Брестского политехнического института “Современные проблемы математики и вычислительной техники” (Брест, 2003), VIII Республиканской научно-технической конференции студентов и аспирантов “НИРС-2003” (Минск, 2003), II и IV Международных научно-методических конференциях “Дистанционное обучение – образовательная среда XXI века” (Минск, 2002; Минск, 2004), VII Республиканской конференции студентов и аспирантов Гомельского государственного университета “Новые математические методы и компьютерные технологии в проектировании, производстве и научных исследованиях” (Гомель, 2004), III Международной научно-технической конференции “Проблемы проектирования и производства радиоэлектронных средств” (Новополоцк, 2004), VI и VII Международной летней школе-семинаре студентов и аспирантов “Современные информационные технологии” (Браслав, 2003; Браслав, 2004), двух конференциях студентов, магистрантов и аспирантов Белорусского государственного университета информатики и радиоэлектроники.

**Опубликованность результатов.** По теме диссертационной работы опубликовано тринадцать работ (общий объем – около 44 стр.), в их числе:

три статьи в научно-теоретическом журнале;

десять тезисов и текстов докладов в сборниках трудов и материалов конференций.

**Структура и объем диссертации.** Диссертация состоит из введения, общей характеристики, четырех глав, заключения, списка литературы и приложений. Работа изложена на 163 страницах машинописного текста, в том числе основная часть – на 113 страницах, и содержит:

32 рисунка;

список использованных источников, включающий 135 наименований и размещенный на 10 страницах;

5 приложений объемом 40 листов.

## ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении приведена краткая оценка современного состояния проблемы построения модельных автоматизированных систем управления сложными производственными объектами, указаны основные исходные предпосылки для разработки темы, обоснована необходимость проведения исследования.

В первой главе рассмотрены вопросы формального обоснования и построения паттернов объектно-ориентированного проектирования и программирования (ООП) управляющей части систем управления с императивным характером определения поведения координируемых элементов.

Концептуальная схема моделей процессов императивного управления дискретным процессом, как известно, может быть представлена с выделением уровней планирования и реализации плана (рис. 1) [12].

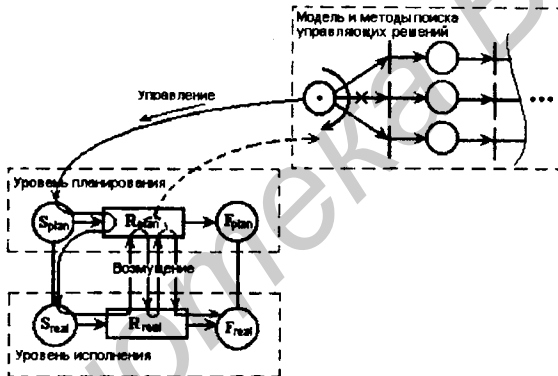


Рис. 1. Схема императивного управления процессом дискретного типа

Поведение системы на уровне планирования рассматривается на состояниях  $S_{plan}$  и  $F_{plan}$ , но на уровне исполнения они проецируются на состояния  $S_{real}$  и  $F_{real}$  соответственно.

Реализуя контроль соответствия между запланированным  $R_{plan}$  и текущим  $R_{real}$  ходом процесса, система управления находится в состоянии обнаружения возмущения  $E_{pt}$ . Однако конечно-автоматный стиль представления системы, часто достаточный для обсуждения схем реализации вычислительного процесса, менее конструктивен для построения процедур поиска управления (рис. 1). Соответствие  $R_{plan} \leftrightarrow R_{real}$  необходимо учитывать, например, для многошаговых процессов с интерактивным назначением целевых состояний.

В качестве атомарных элементов моделей рассматриваемых дискретных процессов выбраны переходные системы вида "условие – действие", определяемые тройками множеств  $S_c$ ,  $S_c = \langle V_c, P_c, A_c \rangle$ , где  $V_c$  – переменные

состояния;  $P_c$  – условия на  $V_c$ , задаваемые вычислимыми предикатами;  $A_c$  – действия, биективно соответствующих условиям  $P_c$ .

Реализация принципа детализации и конкретизации описания в технологии ООП, как известно, основана на модулях-функциях языка программирования. Определяя базовые классы в корне иерархии, удобно использовать расширения двух известных вариантов построения аппарата описания – временных сети Петри с задержками в переходах *TPN (Timed Petri Nets)* и систем переходов, определяемых средствами языка C++.

Показано, что для представления как однородных, так и иерархических и вложенных систем переходов достаточно использовать пятерку функций  $S_c = (E_i, C_i, D_i, F_i, I_i)$ , где  $E_i$  – условия активизации перехода;  $C_i$  – действия, вызываемые в системе при входе перехода в активное состояние;  $D_i$  – длительность во времени активной фазы;  $F_i$  – действие перехода при выходе из активного состояния;  $I_i$  – действия над переходом при внешнем прерывании. Индекс  $i$  соответствует экземпляру перехода.

Шаблон базового класса представления перехода системы  $S_c$ , пригодный для реализации модели типа *PME*, имеет вид:

```
template <class Time> struct Transition {
    int E_next; // Ссылка на следующий активизированный переход
    Time E_time; // Момент пассивизации перехода
    virtual int E(int) = 0; // Условия активизации перехода
    virtual void C(int) = 0; // Действия при активизации перехода
    virtual Time D(int) = 0; // Длительность активной фазы
    virtual void F(int) = 0; // Действие при выходе из активного состояния
    virtual void I(int) {} // Обработка внешних прерываний перехода
};
```

Процесс активизации отдельного перехода, подобно переходам сетей Петри, естественно связать с его восприимчивостью к изменению локальных переменных состояния  $V_i$ :  $V_i = \text{dom}(E_i) \cup \text{dom}(C_i) \cup \text{dom}(F_i) \cup \text{dom}(I_i)$ . Здесь  $\text{dom}(f)$  – множество переменных состояния, связываемых функцией  $f$ . Такое множество образуют глобальные переменные модуля программы, реализующего функцию  $f$ , а в случае, когда  $f$  – функция-элемент класса,  $\text{dom}(f)$  может включать элементы данных класса.

Активная фаза использования модулей-функций языков процедурного типа является элементарным автоматным переходом. На переходах системы  $S_c$  можно формально построить однодольную сеть *IPN*,  $IPN = (A, B, V)$ , связывающую переходы  $A$  с потенциальной возможностью активизации. Здесь  $V$  – переменные состояния – аналог позиций, а  $V = (A \times A) \rightarrow \{0, 1\}$ .

Обозначим  $x$  и  $x'$  – множество входных и выходных элементов любой вершины  $x$  ориентированного графа, соответственно.

Структура смежности графа сети *IPN* в виде *FSF (Forward Star Form)* определяется так:

$$\{a : a'\} = \{a : \{x \mid \text{dom}(E_a) \cap (\text{dom}(F_x) \cup \text{dom}(I_x)) \neq \emptyset, x \in A\}, a \in A\}.$$



Определение графа сети  $IPN$  использовано как для оптимизации управления, так и имитационного моделирования активности переходов.

Рекуррентная схема интерпретации процессов на сети  $IPN$  построена на основе понятий виртуального стартового перехода  $s$  и виртуального финишного перехода  $f$ , для которых справедливо

$$\begin{cases} s' = \emptyset, s = \{a \mid \text{dom}(E_a) \cap \text{dom}(F_s) \neq \emptyset\}, D_s = 0; \\ f' = \emptyset, D_f = 0; \\ A \leftarrow A \cup \{s\} \cup \{f\}. \end{cases}$$

Изменение состояния  $IPN$  при этом оказывается однозначно привязанным к моментам пассивизации переходов. Это влечет не только эффективную реализацию волновой схемы учета последствий изменения состояния сети, но и позволяет рассматривать внешние события в реальном времени как переходы специального вида. Состояние сети на любом этапе  $k$  представляется списком внутренних событий  $I_k$ , элементы которого – момент времени  $t_k$  и номер перехода  $a \in A$ :

$$\begin{cases} I_0 = \{(0, s)\}; \\ I_k = \{(t_i, a), i > k-1, a \in A\}, k > 0. \end{cases}$$

Шаблон класса интерпретатора системы  $S_c$  имеет вид:

```
template <class Time> class S_IPN: public EventScanner<Time> {
    virtual void prolog(); // Пролог интервала активности сети
    virtual void ephilog(); // Пролог интервала активности сети
    virtual void expanded(int i); // Фиксация факта активизации перехода i
protected:
    Transition<Time> *F_tran; // Описание переходов сети
    int *L_tran; // Поле данных списов выходных дуг переходов
    int *M_tran; // Указатели списков выходных дуг переходов

    virtual void expans(int i); // Попытка активизации перехода i
public:
    S_IPN(Transition<Time> *F, int N, int *L, int *M, int I):
        EventScanner<Time>(N, I), F_tran(F), L_tran(L), M_tran(M) {}

    void finished(int i); // Обработка последствий пассивизации перехода i
    ~S_IPN() {}
};
```

Его базовый класс представляет монитор событий, обеспечивающий реализацию рекуррентной схемы интерпретации процессов на сети. Вычислительная сложность обработки этапа пассивизации отдельного перехода  $x$  оценивается величиной  $O(|x| \cdot |x'|)$ ,  $x \in A$ .

Показано, что построенный подобным образом полиморфный класс интерпретации системы переходов обладает не только возможностями

полиморфизма операторов изменения разметки сети, но позволяет задать закон управления последовательностью событий на описании графа  $IPN$ . Рассмотрен способ задания такого закона и для  $IPN$  посредством детализации графа связей переходов на основе  $dom(E_i) \cup dom(C_i) \cup dom(F_i)$ .

Учитывая связь интервалов существования объектов классов ООП с процессами управления памятью, стандартно планируемыми компиляторами языка C++, показано, что объем памяти для размещения статического описания  $IPN$  составляет

$$S_{IPN} = 1 + 3|A| + \sum_{a \in A} |a'|.$$

Это меньше объема памяти описания эквивалентных сетей расширенных  $TPN$ :

$$S_{TPN} = 4 + 5|A| + 2(|B| + \sum_{a \in A} |a'|) + 3 \sum_{a \in A} |a|.$$

Интерпретация процессов на сетях интересна для решения задач, связанных с контролем формальных свойств сети, например, живости, ограниченности и достижимости. В случае неоднородных сетей переходов этот метод единственный. Однако в контуре управления (рис. 1) сеть используется как для поиска решений, так и обнаружения возмущений запланированных траекторий посредством расширения набора продукционных правил.

**Во второй главе** рассмотрены задачи построения шаблонов классов оптимизации управления на сетях переходов. Уровень базовых классов при этом соответствует задачам поиска кратчайших путей на графах. В отличие от классических задач такого вида, здесь предполагается разделение моделей структуры сети и процесса перемещения.

Пусть транспортная сеть, как модель пространства поиска, представлена нагруженным ориентированным графом  $G(M, N)$ , где  $N$  и  $M$  – множества вершин и дуг графа, а каждой дуге  $(i, j) \in M$ ,  $i, j \in N$ , соответствует положительное вещественное число  $w(i, j) < \infty$ , называемое длиной дуги.

С целью отражения условий перемещения отдельной подвижной единицы обозначим [2]:

$cont(i, j)$  – множество допустимых вершин для развития путей из вершины  $j$  для любой дуги  $(i, j) \in M$ ;

$x'$  – множество непосредственно достижимых смежных вершин из вершины  $x$ ,  $x' = \{k \mid w(x, k) \geq 0\}$ .

Очевидно, что  $cont(i, j) \subseteq j'$ , а в случае отсутствия ограничений на выбор пути после прохождения дуги  $(i, j)$  –  $cont(i, j) = j'$ ,  $i, j \in N$ ,  $(i, j) \in M$ .

Пусть вершины  $s$  и  $t$  соответствуют состояниям  $S_{plan}$  и  $F_{plan}$  (рис.1).

Типичный запрос в системе императивного управления, находящейся в состоянии  $S_{plan}$ : найти, если существует, на множестве вершин  $\{s(0)=s, s(1) \in cont(s, s), s(2) \in cont(s(0), s(1)), \dots, s(i) \in cont(s(i-2), s(i-1)), \dots, t\}$  кратчайший путь от вершины  $s$  к вершине  $t$ , где  $s, t \in N$ .

Показано, что непосредственное применение алгоритма Дейкстры, как наиболее эффективного среди известных для задач подобного вида, приводит к неверным результатам (рис. 2) [2,7].

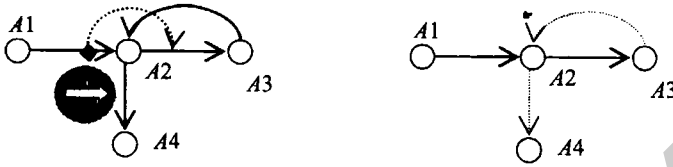


Рис. 2. Пример некорректного применения алгоритма Дейкстры.

При наличии запрета пути  $A1 \rightarrow A2 \rightarrow A4$  (а) дерево путей  $A1 \rightarrow A4$  прервано в  $A3$  (б)

Однако можно сохранить схему Дейкстры с ее высокой вычислительной эффективностью, если переопределить исходный граф (рис. 3).

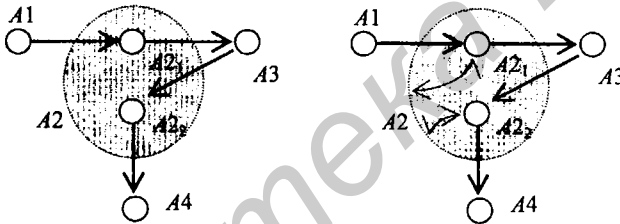


Рис. 3. Учет ограничений расширением модели сети.

Запрет пути  $A1 \rightarrow A2 \rightarrow A4$  (а) отражен введением вершин  $A2_1$  и  $A2_2$ , в результате дерево путей  $A1 \rightarrow A4$  построено правильно (б)

Алгоритм переопределения в общем случае:

для каждой дуги  $(i, j)$  с непустым ограничением, когда  $cont(i, j) \neq j'$ , выполнить замену дуг  $\{(x, j), (x \in j') \wedge (x \neq i)\}$  на дуги  $\{(i, j_k) \cup (j_k, x), x \in j'\}$ , где  $k$  – номер экземпляра дополнительной вершины – копии вершины  $j$ , вводимой для перехода к задаче поиска кратчайших путей без ограничений.

Очевидно, что  $k < |j'|$ , так как вершину  $j$  можно нумеровать нулевым индексом. При расширении подобным образом графа количество вершин увеличивается на

$$\Delta |N| = \sum_{(x,y) \subset M} (cont(x,y) \neq y') |y| - 1,$$

а количество дуг увеличивается на величину

$$\Delta |M| = \sum_{y \in N} \left( \sum_{x \in y} |cont(x,y)| \right) - |y'|.$$

Неудобства реализации рассмотренного приема становятся практически неприемлемыми, например, в многозадачной серверной среде обслуживания запросов, параметризованных относительно вида ограничений.

Предложен способ поиска кратчайших путей на графе с ограничениями, не требующий переопределения графа. Достаточно организовать волновой процесс на очереди дуг, что соответствует моделированию  $TPN$ , задержки в переходах которой соответствуют весам дуг [7].

При регулярном поиске деревьев кратчайших путей на фиксированном графе  $G(N, M)$  можно сократить время обработки запросов сервером:

- исключить этапы многократной инициализации комплекта расстояний;
- уменьшить количество холостых проверок элементов очереди вершин;
- ускорить момент принятия решения на включение вершины в дерево.

Пусть в потоке запросов к серверу выделено множество исходных вершин  $S \subseteq N$  и  $|S|=m$ . Так как отдельное построенное дерево кратчайших путей представимо комплектами расстояний  $R$  и номеров предшествующих вершин  $P$ ,  $|R| = |P| = |N| = n$ , то для расчета  $m$  кратчайших путей потребуется расширение структуры данных представления графа, например, в виде  $FSF$ .

Для сохранения схемы выборки вершин образуем  $m$  виртуальных графов  $G_0, G_1, G_2, \dots, G_{m-1}$ , изоморфных исходному графу  $G$ . Каждый из графов  $G_i$  будет отображен на область данных представления графа  $G$  посредством нумерации вершин по правилу  $x^i = x + i \cdot k$ ,  $x = \overline{0, n-1}$ ,  $i = \overline{0, m-1}$ . Здесь  $x^i$  – номер вершины виртуального графа  $G_i$ , соответствующей вершине  $x$  исходного графа  $G$ ,  $i = \overline{0, m-1}$ ;  $k$  – диапазон изменения номеров вершин исходного графа,  $k \geq n$ . Значение  $k$  целесообразно назначить из соображений эффективной реализации смещения:  $k = 2((n+1)/2)$ , где деление выполнено с отбрасыванием остатка. Обратное отображение в этом случае реализуется операцией сдвига на  $\log_2 k$  позиций.

Построение дерева кратчайших путей, как известно, включает инициализацию комплекта  $R = R(i) = \{R_\infty, i \in N\}$ , где  $R_\infty$  – максимально возможное значение, определяемое выбранным типом данных.

Нетрудно заметить, что процесс построения дерева путей можно начать с любого значения  $R(s_1)$  [7,9]. Действительные значения расстояний восстанавливаются преобразованием  $R(j) \leftarrow R(j) - R(s_1)$ ,  $j \in N$ , предполагая истинность условия  $(R_\infty \gg D) \wedge (R(j) < D)$ . Отсюда следует, что если для каждой вершины  $s_i$  начало отсчета расстояний выбирать по правилу  $R(s_i) = R_\infty - i \cdot D$ ,  $i = \overline{1, m}$ , то отпадает необходимость выполнения этапа инициализации комплекта  $R$  при условии, что  $(i) \bmod (2 \cdot R_\infty / D - 1) > 0$ . Невыполнение такого условия означает переполнение машинного слова представления  $R$ . Оценка сокращения вычислительной трудоемкости построения множества путей –  $O(|S| \cdot |N| \cdot (1 - D/2 \cdot R_\infty))$ . Значение  $D$  может быть оценено предварительно для любого конечного нагруженного графа:  $D \leq \sum_{i \in N} \max_j w(i, j)$ .

Наконец, так как любая часть кратчайшего пути является кратчайшим путем, то для каждой вершины графа, представленного на сервере, возможно предварительное однократное выделение входных дуг минимальной длины:

$$T_j = \left\{ (i, j) : i = \arg \min_{i, j} \{w(i, j) : (i, j) \in M\} \right\}, j \in N.$$

Очевидно, что в случае ветвления процесса через подобную дугу значение расстояния до ее конечной вершины не изменится. Такую вершину можно не включать в очередь, а сделать еще один шаг вперед. Как следствие, количество операций над очередью сокращается. Эксперименты подтверждают повышение быстродействия на 20...40% для  $n \approx 10^5$  и  $|x^i| \leq 10$ ,  $x \in N$ .

Установлено, что наибольший эффект от использования сервером предопределенных решений дает построение иерархии укрупненных моделей графа, когда ограничение пространства поиска на нижнем уровне приводит к сокращению вычислительной сложности  $O(n^2)$  до  $O(n^{5/4})$  [3,9].

**В третьей главе** рассматривались вопросы практической реализации процедур выбора оптимальных вариантов по схеме кооперативного использования ресурсов распределенной вычислительной среды.

Известно, что конкретную задачу выбора  $Z$  можно характеризовать тройкой  $Z = (V, P, S)$ , где  $V$  – множество вариантов, подлежащих оценке по заданному критерию;  $P$  – процедура получения оценки качества для отдельного варианта из множества  $V$ ;  $S$  – процедура реализации вычислительной схемы решения  $Z$ , определяющая порядок применения  $P$  к элементам множества  $V$ .

Предположим, что решается задача минимизации целевой функции  $F$ . Кроме того, используемая для оценки отдельного варианта процедура  $P$  обладает возможностью получения в процессе работы нижних оценок значения функции  $F$ , а  $R$  – текущее значение рекордной оценки, является общедоступным для всех агентов кооперативной системы.

Объектно-ориентированное представление функциональной части агента шаблоном класса на языке C++ может быть задано так:

```
template <class Criterial, class Variant> class Agent {
    volatile Criterial R; // Глобальный объект рекордной оценки
protected:
    virtual Criterial FirstStep() = 0;
    virtual bool NextStep(Criterial &) = 0;
public:
    virtual void P(Variant V) {
        Criterial F = V.FirstStep(); // Предварительная оценка варианта
        if (R > F) { // Организация анализа варианта
            while (V.NextStep(F)) if (R < F) return;
            if (R > F) R = F; // Установление нового рекорда
        }
    }
};
```

Обозначим  $\bar{t}(x)$  – среднее время решения задачи на  $x$  ЭВМ. Известно, что относительный эффект распределения процесса между  $m$  агентами с обменом

рекордными оценками, составляет  $\overline{t(1)/t(m)} = 3m^2/(2m+1)$ . Последнее выражение служит доводом в пользу кооперирования ЭВМ.

Полиморфизм функций сравнения позволяет считать класс Agent проблемно независимым. Система взаимодействующих агентов обеспечивает объединение ресурсов вычислительной сети с целью гарантированного решения исходной задачи  $Z$ . Агенты должны быть размещены во всех потенциально доступных узлах вычислительной сети.

Предложена иерархия классов создания виртуального глобального объекта рекордной оценки на сети, использующая групповое вещание по протоколу UDP.

На примере задачи составления расписания учебных занятий показано, что одной из практических проблем рекурсивных процедур исчерпывающего поиска является недостаток памяти. Для древовидных пространств определения вариантов, однако, можно использовать распределение памяти между кооперируемыми ЭВМ. Например, при реализации метода ветвей и границ узел дерева часто содержит достаточно много информации, чтобы организовать рекурсивный переход к другим узлам с разностной схемой передачи данных[5].

Пусть  $S_i$  и  $C_i$  – множества уникальных кортежей базы данных представления  $V$  парой ЭВМ, играющих на некотором этапе решения задачи  $Z$  роль сервера и клиента. Если базы данных задачи  $Z$  синхронизированы на этапе  $i$ , то  $S_i \equiv C_i$ , но на этапе  $i+1$  возникает потребность передачи сервером изменений текущей копии базы клиента. На уровне теоретико-множественных операций база клиента должна соответствовать

$$C_{i+1} = (C_i \setminus \{S_i \setminus S_{i+1}\}) + \{S_{i+1} \setminus S_i\}.$$

Однако переход  $S_i \rightarrow S_{i+1}$ , в общем случае, не фиксируется на логическом или физическом уровнях стандартной модели данных. Возникающая для его реализации потребность наличия одновременно  $S_i$  и  $S_{i+1}$  в случае относительно незначительных изменений базы сервера, когда  $|S_i \cap S_{i+1}|$  значительно больше  $|S_i \setminus S_{i+1}|$  или  $|S_{i+1} \setminus S_i|$ , делает непривлекательным хранение сервером копии  $S_i$  до этапа  $i+1$ .

Ключом разрешения проблемы является дополнение описания вариантов значениями временных границ интервала  $[A, B]$  существования данных. Если клиент нуждается в базе данных в момент  $t_{i+1}$ , то серверу необходимо лишь получение от клиента значения момента  $t_i$  для определения множеств

$$S_i = \{S \mid (A \leq t_i) \wedge (A \leq B)\} = \{S \mid (t_i \leq B)\},$$

$$S_{i+1} = \{S \mid (A \leq t_{i+1}) \wedge (A > B)\} = \{S \mid (A > B)\}.$$

Ряд вычислительно сложных задач (например, квадратичная задача назначения, задача размещения) требуют выбора оптимальной перестановки или сочетания элементов множества.

Предложен алгоритм порождения в лексикографическом порядке любого из  $k$  подмножеств перестановок из  $n$ , указанного номером  $i$ ,  $i = 0, k-1$  [8,11]. Установка начальных условий процесса порождения имеет вычислительную

сложность  $O(n)$ . В результате для координации процессов анализа вариантов агентами на сети достаточно пронумеровать ее узлы. Передача элементов перестановок и холостой пропуск неиспользуемых перестановок не требуется.

**В четвертой главе** рассмотрено применение полиморфных сетей для построения слоев многоуровневой защиты разрабатываемых на языках программирования C++ программ в вычислительных средах операционных систем семейства Windows. Метод реализации защиты – конструирование объектно-ориентированной оболочки кода программы (функций и данных), позволяющей контролировать условия целостности и безопасности от момента порождения процесса (рис. 4) [1,6].

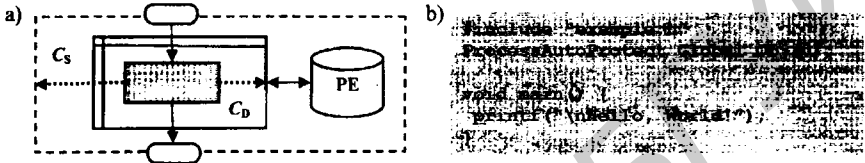


Рис. 4. Схема связи слоев статической ( $C_S$ ) и динамической ( $C_D$ ) защиты исполняемых файлов (а) и пример оформления исходного текста для их установки (б)

Открытость форматов хранения и доступность копирования загрузочных модулей создает угрозы их несанкционированного использования для непосредственного исполнения запрещенных действий, анализа кода программы после дизассемблирования и трассировки программы. Предложена схема автоматического скрытия и контроля использования кода программ, начиная от этапа трансляции исходного текста (рис.5).

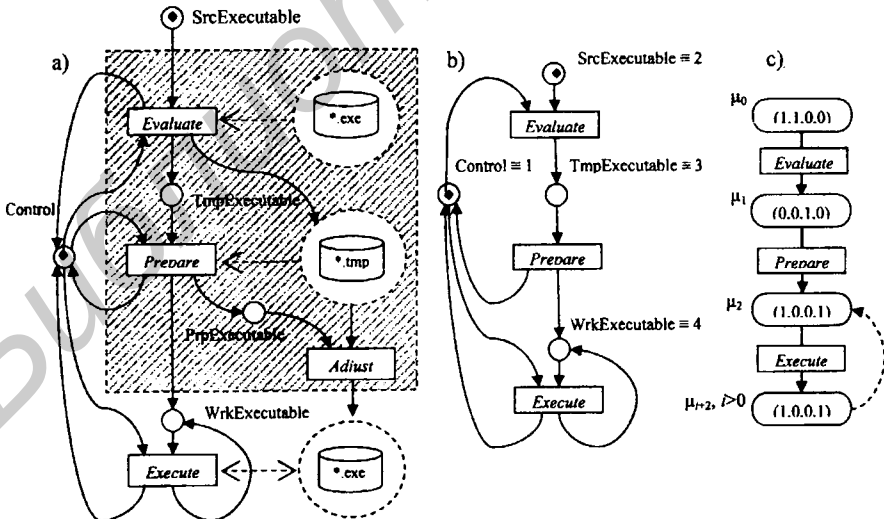


Рис. 5. Сетевая модель статической защиты исполняемых файлов. Для исходной версии (а) эквивалентная сеть Петри (б) имеет дерево достижимости (с) с гарантированно рекурсивным правилом порождения.

Известно, что до вызова головной функции программы или функции-диспетчера динамически подключаемой библиотеки могут вызываться конструкторы глобальных статических объектов. Перед вызовом конструкторов таких объектов файл загрузочного модуля спроецирован полностью на адресное пространство прикладного процесса. Именно подобное стандартное раскручивание процесса исполнения программ использовано для организации перевода кода программы из скрытого состояния в открытое [6].

Наличие описания *IPN* и ее интерпретатора допускает построение многоуровневой системы защиты, где отдельному уровню соответствует шаг рекурсии процесса раскрытия кода программы. Обоснованием надежности защиты при этом является анализ дерева достижимости эквивалентной сети Петри (рис. 5,6).

Связь между уровнями требует учета логики работы программы. Первый уровень, соответствующий переходу от скрытой версии кода и данных в файле загрузочного модуля, назван уровнем статической защиты  $C_S$  (рис.4).

Статическая защита преследует цель предотвращения раскрытия кода программы и данных после считывания всего файла программы в память с целью дизассемблирования. Вместе с тем, потребность в защите любого фрагмента кода существует до момента его непосредственного исполнения.

Для отдельных функций или даже операторных блоков программы предлагается размещать агенты защиты кода, оформленными как объекты классов с функциональным замыканием процедуры защиты конструкторами и деструкторами. Агенты существуют на интервале исполнения защищаемого блока кода. Раскрытие кода тела функции или операторного блока, таким образом, может быть отложено на более поздний этап выполнения программы. Отражая возможность многократной активизации защищаемых областей кода, а также потенциально допустимую вложенность операторных блоков, образуемые уровни названы уровнями динамической защиты (рис. 6).

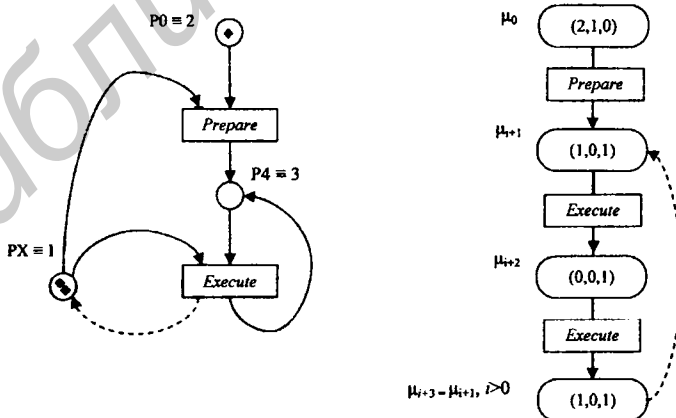


Рис. 6. Сеть Петри системы динамической защиты кода и данных и дерево ее достижимости



С целью скрывтия фаз процесса функционирования системы защиты (рис. 5а), предложен алгоритм контроля и установки масок доступа к процессам и потокам, гарантирующий безопасность смены состояний программы [1].

В заключении сформулированы основные результаты работы.

В приложениях приведены исходные тексты программ с контрольными примерами, а также документы, подтверждающие внедрение результатов диссертационной работы.

## ЗАКЛЮЧЕНИЕ

Результатом диссертационной работы явилось развитие открытых для расширения и детализации формальных схем описания многоуровневых дискретных взаимодействующих процессов и создание новых средств их алгоритмической интерпретации в рамках методов объектно-ориентированного моделирования и проектирования, допускающих построение паттернов проектирования систем с императивным характером управления, что позволяет воспользоваться преимуществами современных технологий объектно-ориентированного программирования.

Основные научные и практические результаты исследований:

1. Предложено уточнение аппарата описания дискретных процессов в системах с императивным характером поведения, ориентированные на объектно-ориентированный стиль спецификации контура управления [5,6]. Полиморфные сетевые модели обеспечивают технологическое объединение фаз построения системы управления – формализацию, алгоритмизацию и программную реализацию на основе конкретизации организационных и технологических требований в рамках формальных систем продукций, определяемых непосредственно языком программирования.

2. Исследованы возможности эффективной реализации принципа конкретизации описания на основе полиморфных расширений сетей Петри для неоднородных переходных систем общего вида. Построены интерпретаторы таких систем в форме параметризованных шаблонов классов [6], в которых особые состояния отображения “система-процесс” предложено совмещать с процессами поиска решений [10,12].

3. Разработаны эффективные по памяти и быстрдействию алгоритмы поиска кратчайших путей на графах абстрактных транспортных сетей с ограничениями [4], учитывающих реальные технологические и временные ограничения и обеспечивающие быстореактивную обработку потока запросов на выборку путей в реальном времени [2,3,7].

4. Предложен вариант специализации операторов присваивания шаблона базового класса агента решения задач выбора оптимальных вариантов на распределенных вычислительных средах [13], позволяющий реализовать

принцип накопления опыта [9,10] посредством кооперации вычислительных ресурсов сети.

5. Разработан алгоритм генерации подмножеств перестановок независимых вариантов [5,11] с линейной сложностью установки начального состояния [8]. Для случая зависимых вариантов в процедурах типа ветвей и границ построены не требующие дублирования данных схемы синхронизации памяти кэша агентов [9].

6. Разработаны базовые классы обеспечения безопасности работы [6] агента кооперативных вычислений, основанные на создании слоев статической и динамической схем контроля условий целостности и безопасности. Предложен способ организации системы контроля в реальном времени масок доступа к объектам системы [1].

Результаты исследований практически использованы при выполнении НИОКР для модернизации систем управления установками компаундирования масел на Новополоцком нефтеперерабатывающем заводе “Нафтан”, а также для обоснования схем активного противодействия угрозам безопасности в прикладных системах обработки информации ответственного назначения, что подтверждается актами внедрения.

## СПИСОК ОПУБЛИКОВАННЫХ РАБОТ ПО ТЕМЕ ДИССЕРТАЦИИ

### Статьи:

1. Ревотюк М.П., Колотыгин К.Е., Дарадкех Ю.И. Контроль защиты потоков разрабатываемых программ в среде Windows NT/2000//Известия Белорусской инженерной академии, № 1(15)2, 2003. – С. 248-250.
2. Ревотюк М.П., Дарадкех Ю.И., Кирейчук В.А. Поиск кратчайших путей на графах полиморфных сетей с ограничениями//Известия Белорусской инженерной академии, № 1(17)1, 2004. – С. 126-129.
3. Ревотюк М.П., Колотыгин К.Е., Дарадкех Ю.И.. Ускорение процесса регулярного поиска кратчайших путей на графах//Известия Белорусской инженерной академии, № 1(17)3, 2004. – С. 87-90.

### Тезисы и тексты докладов в сборниках и материалах конференций:

4. Дарадкех Ю.И., Ревотюк М.П. Поиск кратчайших путей на графах при узловых ограничениях на структуру пути//Материалы III Респ. научной конф. молодых ученых и студентов (Брест, 26-28 ноября 2003 г.) – Брест: БГТУ, 2003. – С. 86-90.
5. Ревотюк М.П., Дарадкех Ю.И., Колотыгин К.Е. Оптимизация расписания учебных занятий на полиморфных сетевых моделях//Дистанционное обучение – образовательная среда XXI века/Материалы II Междунар.

- научно-метод. конф. (Минск, 26-28 ноября 2002 г.) – Мн.: Бестпринт, 2002. – С. 421-423.
6. Ревотюк М.П., Дарадкех Ю.И. Объектная схема многоуровневой статической защиты разрабатываемых программ на языках C/C++// Информационные системы и технологии(IST'2002): Материалы I Междунар. конф. (Минск, 5-8 ноября 2002 г.): В 2 ч. Ч.2 – Мн.: БГУ, 2002. – С. 84-89.
  7. Дарадкех Ю.И., Колотыгин К.Е., Ревотюк М.П. Полиморфный класс оптимизации путей на графах//Новые математические методы и компьютерные технологии в проектировании, производстве и научных исследованиях: Материалы VII Респ. научной конф. студентов и аспирантов (Гомель, 22-24 марта 2004 г.). – Гомель: ГГУ, 2004.– С.260-261.
  8. Кишкевич А.П., Дарадкех Ю.И., Ревотюк М.П. Порождение подмножеств перестановок//Новые математические методы и компьютерные технологии в проектировании, производстве и научных исследованиях: Материалы VII Респ. научной конф. студентов и аспирантов (Гомель, 22-24 марта 2004 г.) – Гомель: ГГУ, 2004.– С.213-214.
  9. Ревотюк М.П., Колотыгин К.Е., Дарадкех Ю.И. Иерархическая декомпозиция задач поиска кратчайших путей на полиморфных сетях // Проблемы проектирования и производства радиоэлектронных средств: Сб. материалов III Междунар. научно-технической конф. В 2-х томах. Том 2. – Новополоцк: ПГУ, 2004. – С.135-138.
  10. Ревотюк М.П., Дарадкех Ю.И. Использование предопределенных решений алгоритмами регулярного поиска кратчайших путей //Информационные системы и технологии (IST'2004): Материалы II Междунар. конф. (Минск, 8-10 ноября 2004 г.): В 2.ч. Ч.2. – Мн.: Академия управления при Президенте Республики Беларусь, 2004. – С.127-132.
  11. Ревотюк М.П., Кишкевич А.П., Дарадкех Ю.И. Генерация подмножеств перестановок вариантов//Дистанционное обучение – образовательная среда XXI века: Материалы IV Междунар. научно-метод. конф. (Минск, 10-12 ноября 2004 г.) – Мн.: БГУИР, 2004. – С.276-279.
  12. Дарадкех Ю.И., Колотыгин К.Е., Ревотюк М.П. Модели координации взаимодействующих процессов на основе полиморфных сетей//“НИРС-2003”. VIII Респ. научно-технич. конф. студентов и аспирантов (Минск, 9-10 декабря 2003 г.)/Тезисы докладов. – Мн.: БНТУ, 2003. – С.72.
  13. Кузнецова Н.В., Дарадкех Ю.И., Ревотюк М.П. Управление потоками вариантов в кооперативных схемах решения задач выбора//“НИРС-2003”. VIII Респ. научно-технич. конф. студентов и аспирантов (Минск, 9-10 декабря 2003 г.)/Тезисы докладов. – Мн.: БНТУ, 2003. – С.74.



## РЭЗЬЮМЭ

дысертацыйнай работы Ю.І.Ю. Дарадкеха

«Інтэрпрэтацыя паліморфных сеткавых мадэляў ў размеркаваных сістэмах рэальнага часу»

**Ключавыя словы:** аб'ектная спецыфікацыя контура кіравання, пераходныя сістэмы, паліморфныя расшыраныя сеткі Петры, шаблоны класаў інтэрпрытацыі працэсаў на сеткавых структурах.

Аб'ект даследавання – праграмае забеспячэнне сістэм кіравання аб'ектамі з дыскретным характарам паводзін і тэхналогія яго распрацоўкі. Предмет даследавання – метады рэалізацыі працэсаў пошуку аптымальных рашэнняў ў рэальным часе ў адкрытых на узроўні мэты кіравання узаемадзейнічаючых сістэмах. Мэта работы – распрацоўка на аснове іерархіі шаблонаў поліморфных класаў сродкаў канкрэтызуемай спецыфікацыі праграмага забеспячэння сістэм кіравання дыскрэтнымі працэсамі.

Прапанавана развіццё адкрытых для пашырэння і дэталізацыі схем спецыфікацыі дыскрэтных працэсаў, апіраючыхся на канцэпцыю рэкурсіўна пашыраемых сетак Петры і арыентаваных на рэалізацыю ў сістэмах хуткай распрацоўкі праграм ў рамках кампанентнай мадэлі *Property-Method-Event*. У адрозненне ад вядомых пашырэнняў сетак Петры, зняты абмежаванні на выкарыстанне лексічных элементаў языка C++ пры дэкларацыі ўкладзеных і перарываемых пераходаў, што дазволіла адкрыць шлях эфектыўнага праграмавання працэдур кіравання ў рэальным часе.

Распрацаваны эфектыўныя па памяці і хуткасці алгарытмы аптымізацыі кіравання для задач, звязаных да пошуку найкарачэйшых шляхоў на віртуальных графах з абмежаваннямі на структуры шляхоў. Даследаваны прыемы павышэння хуткасці сервернай апрацоўкі запытаў, дазваляючых за кошт уліку парярэдніх рашэнняў знізіць парадак вылічальнай складанасці.

Прапанаваны метады падзелу і сінхранізацыі дадзеных падмностваў сямействаў варыянтаў пры рашэнні задач пошуку аптымальных рашэнняў па схеме кааператыўнага выкарыстання рэсурсаў вылічальных сетак, дазваляючых палепшыць дынамічныя характэрыстыкі працэсу пошуку рашэнняў ў рэальным часе.

Распрацаваны і абгрунтаваны шаблоны класаў многаузровеньнай абароны выконваемых модуляў, дазваляючых забяспечыць патрэбны узровень гарантый цэласнасці і бяспекі размеркаваных вылічальных працэсаў з аўтаматызацыяй тэхналагічных этапаў устаноўкі абароны.

Вынікі дысертацыйнай работы ўкаранены ў радзе праектных устаноў, аб чым маюцца акты ўкаранення.

## РЕЗЮМЕ

диссертационной работы Ю.И.Ю. Дарадкеха

“Интерпретация полиморфных сетевых моделей в распределенных системах реального времени”

Ключевые слова: объектная спецификация контура управления, переходные системы, полиморфные расширенные сети Петри, шаблоны классов интерпретации процессов на сетевых структурах.

Объект исследования – программное обеспечение систем управления объектами с дискретным характером поведения и технология его разработки. Предмет исследования – методы реализации процессов поиска оптимальных решений в реальном времени в открытых на уровне цели управления взаимодействующих системах. Цель работы – разработка на основе иерархии шаблонов полиморфных классов средств конкретизируемой спецификации программного обеспечения систем управления дискретными процессами.

Предложено развитие открытых для расширения и детализации схем спецификации дискретных процессов, опирающихся на концепцию рекурсивно расширяемых сетей Петри и ориентированных на реализацию в системах быстрой разработки приложений в рамках компонентной модели *Property-Method-Event*. В отличие от известных расширений сетей Петри, сняты ограничения на использование лексических элементов языка C++ при декларации вложенных и прерываемых переходов, что позволило открыть пути эффективного кодирования процедур управления в реальном времени.

Разработаны эффективные по памяти и быстродействию алгоритмы оптимизации управления для задач, сводимых к поиску кратчайших путей на виртуальных графах с учетом ограничений на структуры путей. Исследованы приемы повышения быстродействия серверной обработки запросов, позволяющие за счет учета предопределенных решений снизить порядок вычислительной сложности.

Предложены методы разделения и синхронизации данных определения подмножеств семейств вариантов при решении задач поиска оптимальных решений по схеме кооперативного использования ресурсов вычислительных сетей, позволяющие улучшить динамические характеристики процесса поиска решений в реальном времени.

Разработаны и обоснованы шаблоны классов многоуровневой защиты исполняемых модулей, позволяющих обеспечить требуемый уровень гарантий целостности и безопасности распределенных вычислений с автоматизацией технологических этапов установки защиты.

Результаты диссертационной работы внедрены в ряде проектных организаций, о чем имеются акты внедрения.

## SUMMARY

of dissertation work by Y.I.Y. Daradkeh  
“Interpretation of Polymorphous Networks Models in Distributed  
Real-Time Systems”

Key words: object specification of the control circuit, transitive systems, polymorphous extended Petri nets, template classes of the processes interpretation based on network structures.

This work focuses on the development of the methods for the control system software design of the discrete processes based on computer aids using object-oriented modeling and programming technology. The main objective of this work – development on the basis of hierarchy of patterns of polymorphic classes of means of the concretized specification of the software of control systems of discrete processes.

Development opened for expansion and is offered to detailed elaboration of schemes of the specification of the discrete processes leaning the concept of recursively expanded Petri nets and focused on realization in systems of rapid application design of appendices within the limits of componental model Property-Method-Event. Unlike known expansions of Petri nets, restrictions on use of lexical elements of language C++ are removed at the declaration of the enclosed and interrupted transitions that has allowed to open ways of effective coding.

Algorithms of optimization of management are developed for the problems reduced to search of the shortest paths on virtual graphs in view of restrictions on structures of paths effective on memory and speed. Receptions of increase of speed of server processing of the inquiries are investigated, allowing due to the account of the predetermined decisions to lower the order of computing complexity.

Methods of division and synchronization of data of definition of subsets of families of variants are offered at the decision of problems of search of optimum decisions under the scheme of cooperative use of resources of the computer networks, allowing to improve dynamic characteristics of process of search of decisions in real time.

Templates of classes of multilevel protection of the executed modules are developed and proved, allowing to provide a demanded level of guarantees of integrity and safety of the distributed calculations with automation of technological stages of protection installation.

The results of this research are being applied in industry by a set of research organizations.

Юсеф Ибрахим Юсеф  
ДАРАДКЕХ

**ИНТЕРПРЕТАЦИЯ ПОЛИМОРФНЫХ СЕТЕВЫХ МОДЕЛЕЙ  
В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ РЕАЛЬНОГО ВРЕМЕНИ**

05.13.11 – Математическое и программное обеспечение  
вычислительных машин, комплексов и компьютерных сетей

Автореферат диссертации  
на соискание ученой степени кандидата технических наук

---

Подписано в печать	12.12.2005.	Формат 60x84 1/16.	Бумага офсетная.
Гарнитура «Таймс».	Печать ризографическая.		Усл. печ. л. 1,4.
Уч.-изд. л. 1,2.	Тираж 70 экз.		Заказ 772.

---

Издатель и полиграфическое исполнение: Учреждение образования  
«Белорусский государственный университет информатики и радиоэлектроники»  
Лицензия на осуществление издательской деятельности №02330/0056964 от 01.04.2004.  
Лицензия на осуществление полиграфической деятельности №02330/0131518 от 30.04.2004.  
220013, Минск, П. Бровки, 6.