

На правах рукописи

Гришмановский Павел Валерьевич

**Предикатно-матричные сети в задачах анализа и синтеза
систем управления**

Специальность 05.13.01 – «Управление в технических системах»

Автореферат

диссертации на соискание ученой степени
кандидата технических наук

Томск – 2000

Работа выполнена на кафедре Автоматики и Компьютерных Систем
Томского Политехнического Университета.

Научный руководитель –	доктор технических наук, профессор Цапко Геннадий Павлович
Официальные оппоненты -	доктор технических наук, профессор, заведующий отделом вычислительных систем Института физики полупроводников СО РАН Хорошевский Виктор Гаврилович; кандидат технических наук, доцент, начальник департамента по информационным технологиям ОАО «ТомскНИПИнефть ВНК Комагоров Владимир Петрович
Ведущая организация -	Научно-производственное объединение Прикладной механики, г. Железногорск.

Защита диссертации состоится 21 июня 2000 г. в 15 часов на заседании
диссертационного совета Д 063.80.03 в Томском политехническом университете
по адресу: г. Томск, ул. Советская, 84.

С диссертацией можно ознакомиться в библиотеке Томского
Политехнического университета.

Автореферат разослан 20 мая 2000 г.

Ученый секретарь
Диссертационного совета
Кандидат технических наук, доцент

Чудинов И.Л.

Общая характеристика работы

Актуальность проблемы. Системы управления проектируются с целью обеспечения возможности выполнения и решения сложных комплексных задач и обеспечения бесперебойного протекания разнообразных технологических, эксплуатационных и других процессов. Наличие различных функциональных и структурных уровней системы управления позволяет объединить в рамках одного комплекса достаточно разнообразные по физической природе и уровню сложности компоненты в единую среду, функционирующую как неразрывное целое. Современные системы управления характеризуются большим количеством и разнообразием протекающих в них процессов, вследствие чего приобретают черты информационных систем с распределенной обработкой данных. Присущая сложной системе возможность изменения структуры и режимов функционирования приводит к динамическим изменениям условий протекания процессов внутри нее. Как правило, качественная и количественная оценка таких изменений является весьма затруднительной, а зачастую и просто невозможной без проведения модельного эксперимента.

Значительная сложность, важность решаемых задач и, часто, высокая стоимость подобных систем определяют необходимость тщательной их подготовки к выполнению требуемых функций и анализа функционирования еще на этапе проектирования. Кроме того, невозможность обучения и тестирования эксплуатационного персонала на реальных системах приводит к необходимости создания специализированных средств, позволяющих имитировать функционирование этих систем.

Потребность в описанных средствах диктует необходимость разработки не только программно-аппаратных средств исследования систем управления с помощью модельного эксперимента, но и нового формального аппарата, составляющего основу этих средств, для описания сложных систем, характеризующихся наличием параллельно протекающих процессов и возможных структурных и алгоритмических изменений в процессе функционирования. Такой аппарат должен позволять адекватно описывать динамику функционирования как отдельных подсистем, так и системы в целом, и быть свободным от присущих существующим формальным моделям распределенных систем недостатков, в основном таких, как недостаточная описательная и моделирующая мощность, низкая приспособленность для описания реальных систем, ограниченный и фиксированный синтаксис, структурная избыточность, не увеличивающая описательной и функциональной мощности, слабое использование принципов объектно-ориентированного подхода или их отсутствие вовсе. Это является прямой предпосылкой к разработке новых формальных средств описания и анализа систем в соответствии с современным уровнем их представления, отражающих иерархическую природу системы управления в терминах предметной области исследований благодаря применению объектно-ориентированного подхода.

Целью работы является разработка нового формального аппарата имитационного моделирования сложных систем управления, характеризующихся наличием параллельно протекающих процессов и возможных структурных и алгоритмических изменений в процессе

функционирования, исследование и анализ возможностей этого аппарата для моделирования распределенных систем управления и создание на этой основе алгоритмического и программного обеспечения для решения задач создания, отладки, эксплуатации моделей систем управления, проведения модельного эксперимента и организации взаимодействия моделей с программно-аппаратной средой.

Методы исследования. Для решения поставленных задач в диссертационной работе применены методы сетевого имитационного моделирования, теории сетей Петри, теории множеств, теории планирования экспериментов, логики предикатов, матричного исчисления, структурного программирования, объектно-ориентированного проектирования. Проверка основных теоретических положений проводилась экспериментально с использованием как стандартных, так и специально разработанных средств.

Научная новизна работы:

1. Предложен новый объектно-ориентированный формальный аппарат предикатно-матричных сетей (ПМС) для описания и анализа динамики функционирования систем управления, особенностью которого является задание семантики узла, как основного функционального компонента, при помощи комбинации предикатов и матрицы маршрутизации, а динамика модели в целом задается как на уровне компонентов, так и на уровне правил.
2. Применение предикатно-матричного способа задания семантики узлов графа ПМС позволяет при помощи формальных методов производить анализ и интерпретацию моделей систем управления, используя достаточно просто реализуемые операции алгебры логики и матричной арифметики и интуитивно понятные разработчику формы задания семантики компонентов модели.
3. Реализовано свойство реентерабельности узлов графа ПМС в качестве механизма описания совместных непримитивных событий в параллельно протекающих процессах. Данное свойство также отражает реентерабельность программного кода, а его реализация гарантирует при этом отсутствие конфликтов при совместном использовании ресурсов несколькими процессами.
4. Разработаны методы анализа структурных и динамических свойств модели, в том числе достижимости маркировки в ПМС, соответствующей определенным состояниям исследуемой системы.
5. Созданы программные средства имитационного моделирования, обеспечивающие создание, отладку, модификацию и анализ моделей систем управления. В основу данных средств положен разработанный формальный аппарат ПМС.
6. Разработана спецификация открытого программного интерфейса модели, обеспечивающего возможность ее использования, как самостоятельного программного модуля, в составе программно-аппаратного комплекса в качестве алгоритмического ядра системы. Собственно ПМС являются метаязыком и служат для описания взаимодействия подсистем и параллельно протекающих процессов, в то время как для реализации специфических операций используется один или несколько языков программирования высокого уровня.

Практическую ценность работы представляют:

1. Формализованные методики построения моделей с использованием созданного формального аппарата имитационного моделирования, а также методы и алгоритмы анализа получаемых с его помощью моделей.
2. Созданное алгоритмическое и программное обеспечение в виде среды имитационного моделирования, использующей формальный аппарат ПМС и нацеленной на создание моделей систем управления и программного обеспечения тренажерных комплексов и имитаторов.

Реализация результатов работы. Разработанное алгоритмическое и программное обеспечение используется в Научно-производственном объединении Прикладной механики г. Железногорск для исследования отказоустойчивости функционирования бортовых комплексов управления (БКУ) космических спутников связи в процессе их проектирования. Основные положения диссертации внедрены в процесс проектирования БКУ космических аппаратов в НПО ПМ г. Железногорск.

Апробация работы. Результаты диссертационной работы докладывались и обсуждались: на Первом Российско-Корейском международном симпозиуме по науке и технологии KORUS'97, г. Ульсан, Южная Корея, 1997г.; на Третьем Российско-Корейском международном симпозиуме по науке и технологии KORUS'97, г.Новосибирск, Россия, 1999г.; на региональной научно-технической конференции студентов и молодых специалистов «Радиотехнические и информационные системы и устройства» в секции «Автоматизированные системы обработки информации, управления и проектирования», г.Томск, Россия, 1999г.; на V областной научно-практической конференции студентов, аспирантов и молодых ученых «Современные техника и технологии», г. Томск, Россия, 1999г.; на VII Международной научно-методической конференции «Новые информационные технологии в университетском образовании», г. Томск, Россия, 2000 г.; на Четвертом Российско-Корейском международном симпозиуме по науке и технологии KORUS'2000, г. Ульсан, Южная Корея, 2000г.

Публикации. По теме диссертационной работы опубликовано 13 статей и тезисов докладов.

Структура и объем работы. Диссертационная работа состоит из введения, четырех глав и заключения, изложенных на 132 страницах машинописного текста, включает 18 рисунков, 3 таблицы, а также содержит список литературы из 81 наименования и приложения. Общий объем работы - 147 страниц.

Содержание работы

Во введении приводится обоснование актуальности темы, формулируется цель работы, указывается научная новизна и практическая ценность полученных результатов, описывается основное содержание работы.

В первой главе показано, что современные системы управления во многих случаях отличаются сложными структурой и алгоритмами функционирования, оперируют большими объемами информации и представляют собой системы распределенной обработки информации. Тенденции развития современных вычислительных средств и информационных

технологий приводят к постоянному возрастанию сложности объектов управления и, следовательно, систем управления, приобретающих черты информационных систем. В виду сложности объектов управления, характерными чертами современных систем управления являются реализация реального масштаба времени с необходимостью обеспечения непрерывности управления и обеспечение длительного срока функционирования системы управления с высокой степенью надежности. Составляющие ее подсистемы должны, при этом, удовлетворять тем же требованиям, т.е. обладать высокой надежностью и обеспечивать реакцию на входные воздействия за время, не превышающее допустимого времени реакции. Данные условия зачастую являются противоречивыми, так как выполнение одного ведет к нарушению другого. В связи с этим, а также вследствие сложности реализуемого алгоритма, системы управления часто представляют собой распределенные вычислительные программно-аппаратные комплексы, в том числе, состоящие из нескольких ЭВМ.

В распределенных системах имеет место реальный параллелизм событий и процессов, а не виртуальный, как в многозадачных вычислительных системах. Это приводит к сложности формального описания и анализа распределенных систем. Возникает необходимость в теоретическом аппарате, учитывающем данную специфику, что диктует необходимость разработки преимущественно сетевых моделей и методов описания и анализа распределенных систем, таких как сети Петри и др. Сложные системы управления обычно отличаются иерархической структурой и состоят из взаимозависимых подсистем, которые, в свою очередь, также могут быть разделены на подсистемы вплоть до самого низкого уровня. Таким образом, архитектура сложных систем складывается из компонентов и иерархических отношений между этими компонентами, а особенности системы обусловлены отношениями и взаимодействиями между ее частями.

Общие абстракции и механизмы значительно облегчают описание и анализ сложных систем. Это дает возможность выделить структуру классов, структуру объектов и, возможно, структуры модулей и процессов. Причем, каждая из этих структур образуют свою многоуровневую иерархию, в которой элементы более высокого уровня построены из более простых. Структуры классов и объектов не являются независимыми - каждый элемент структуры объектов представляет специфический экземпляр определенного класса, а структуры классов и объектов системы в совокупности образуют архитектуру системы. Исключение структуры классов из описания системы приводит к необходимости повторять одни и те же сведения для каждого однотипного объекта. С введением структуры классов в ней размещаются общие свойства экземпляров.

Внутри любого рассматриваемого уровня находится следующий уровень сложности, поэтому то, какие классы и объекты в рассматриваемой системе выбраны в качестве элементарных, зависит от того, какие результаты ожидаются при исследовании системы, а также от терминологии, принятой в данной предметной области. В любом случае, иерархическая система может быть разделена на четко идентифицируемые части. Такая идентификация возможна, в первую очередь, за счет разделения функций между частями системы, что дает

возможность относительно изолированно изучать каждую ее часть. При описании сложной системы с целью ее анализа или синтеза необходимо путем декомпозиции до приемлемого уровня детализации выявить структуры классов и объектов системы в соответствии с иерархическим представлением этих структур. При этом наиболее эффективным является применение объектно-ориентированного подхода в сочетании со структурным.

Для имитационного моделирования сложных систем управления, характеризующихся одновременностью протекающих в них процессов, широко используются различные сетевые методы, такие как алгоритмические сети, сети Петри и множество модификаций этого математического аппарата. Успех применения методов данного класса заключается в том, что подобное описание распределенной системы является определенной абстракцией самой системы, отражая, при этом, ее структуру - множество подсистем и их взаимосвязей, причем, такое представление является основным в силу интуитивно понятного родства с предметом исследования. Соответствие используемых абстракций конкретным объектам или явлениям определяется предметной областью исследований и предназначением используемой формальной системы, в том случае, если она является специальным аппаратом. Более того, многие из сетевых методов описания сложных систем представляют собой алгоритмические системы, позволяющие применять их для решения общих вычислительных задач при наличии соответствующих программных средств.

По материалам литературных источников проведен анализ различных формальных аппаратов, в том числе сетей высокого уровня, таких как Раскрашенные сети Петри и E-сети. В результате сопоставления возможностей аппаратов имитационного моделирования, сделан вывод о присущих им особенностях и недостатках, делающих в ряде случаев затруднительным их использование для описания реальных систем, а также об отсутствии в данных аппаратах объектно-ориентированных конструкций, что ограничивает их гибкость.

В результате проведенного анализа установлено, что существующие формальные аппараты описания и анализа распределенных систем в основном имеют такие недостатки, как недостаточная описательная и моделирующая мощность, низкая приспособленность для описания реальных систем, ограниченный и фиксированный синтаксис, структурная избыточность, не увеличивающая описательной и функциональной мощности, слабое использование принципов объектно-ориентированного подхода или их отсутствие вовсе. Эти факторы являются прямой предпосылкой к разработке новых формальных средств описания и анализа систем в соответствии с современным уровнем их представления.

В заключение главы сделан вывод о том, что для решения задач анализа и синтеза сложных систем управления, необходим формальный аппарат, который должен обеспечивать представление параллелизма протекающих процессов, непримитивности событий и их одновременности, функциональной и структурной иерархичности системы. Кроме того, необходима гибкость этого аппарата в задании семантики компонентов модели, возможность разделения сложности модели между ее структурой и характеристиками компонентов

модели, легкая процедура адаптация формального аппарата к предметной области исследований, что достижимо при использовании объектно-ориентированного подхода.

Во второй главе для решения поставленных задач предложен новый формальный аппарат предикатно-матричных сетей (ПМС), особенностью которого является задание семантики узла, как основного функционального компонента, при помощи комбинации предикатов и матрицы маршрутизации, полностью определяющих его поведение. Динамика модели задается как на уровне компонентов, так и на уровне правил, что придает формальному аппарату ПМС гибкость при описании сложных систем управления. Структурно ПМС представляют собой ориентированный граф, компоненты которого - узлы и дуги - обладают определенной семантикой, определяющей общие динамические свойства ПМС. С формальной точки зрения, ПМС являются расширением сетей Петри.

Представляемый математический аппарат ПМС и интерпретация его структурных компонентов позволяет рассматривать любую элементарную схему, образованную узлом и окружающими его дугами, как информационную систему, для которой присущи такие фундаментальные системные составляющие, как носители информации, т.е. фишки, структура путей, по которым может осуществляться транспортировка носителей информации, хранение информации, затраты времени, связанные с преобразованием информации и преобразование информации, обеспечиваемые введением специальных процедур, управление транспортировкой, осуществляемое с помощью соответствующих управляющих процедур. Наличие в элементарной схеме, как информационной системе, указанных свойств позволяет эффективно применять формальный аппарат ПМС для моделирования сложных динамических информационно-управляющих систем, для которых характерны все приведенные выше составляющие. При этом, дуги ПМС можно рассматривать как метапеременные сети, а узлы - как метаоператоры, т.е. активные компоненты, способные влиять на состояние метапеременных сети. Метаданными являются фишки - динамические объекты ПМС.

Учитывая, что структура элементарной схемы дополнена правилами выполнения, задающими алгоритм и динамику ПМС в целом, определенную таким образом элементарную сеть можно рассматривать как функциональную систему общего вида, достаточную для описания любой динамической системы.

ПМС формально определена в виде пятерки

$$G = (P, V, T, C, R), \text{ где}$$

P - конечное непустое множество объектов, называемых узлами графа G ; V - конечное множество объектов, называемых дугами графа G ; T - конечное множество объектов, называемых фишками; C - конечное непустое множество классов; R - конечное непустое множество правил. Структура графа G определяется отображением Q :

$$Q: (P \times V \cup V \times P) \rightarrow V \mid V = \{0, 1\}.$$

Маркировкой ПМС называется функция μ , заданная на множестве дуг V и определяющая соответствие каждой дуги подмножеству фишек (в том числе пустому) из множества T , причем любая фишка принадлежит маркировке ровно одной дуги:

$\mu: V \rightarrow T$.

Важной характеристикой является численное выражение маркировки $\eta(v) = |\mu(v)|, \forall v \in V$. Маркировка любого множества дуг $V' \subseteq V$ определяется и как вектор.

Маркировка является динамической характеристикой сети и изменяется на каждом шаге интерпретации ПМС. Маркировка и ее численное выражение на i -ом шаге интерпретации обозначается как μ_i и η_i соответственно. Интерпретация ПМС заключается в последовательном изменении маркировки:

$M = \{\mu_0, \mu_1, \dots, \mu_i, \dots, \mu_n\}$ где

n - последний шаг интерпретации.

Интерпретация ПМС продолжается до тех пор, пока не будет выполнено заданное количество шагов, достигнута заданная маркировка или пока не возникнет тупиковая ситуация. Соответствующее последнему шагу значение модельного времени будет обозначено как τ_n , а весь интервал интерпретации - $[0, \tau_n]$.

На маркировку любой дуги может быть наложено ограничение максимального, минимального, или одновременно максимального и минимального количества фишек в маркировке дуги. Дуга $v \in V$ представлена в виде:

$v = (\eta^L, \eta^H)$, где

η^L - допустимое минимальное количество фишек в маркировке дуги v , $\eta^L \in N_0$;

η^H - допустимое максимальное количество фишек в маркировке дуги v , $\eta^H \in N$.

Таким образом, при интерпретации ПМС всегда истинно условие:

$\eta^L(v) \leq \eta_i(v) \leq \eta^H(v), \forall v \in V, i \in [0, \tau_n]$.

На множестве дуг V определены предикаты β^0 и β^1 , определяющие соответственно возможность добавления и удаления фишки в маркировке дуги $v \in V$, т.е. увеличения и уменьшения ее маркировки:

$\beta^0(v) = (\eta(v) < \eta^H(v))$;

$\beta^1(v) = (\eta(v) > \eta^L(v))$.

Данные предикаты используются для интерпретации сети при определении возможности срабатывания узла. Узел $p \in P$ представляет собой

$p = (E, U, H, r)$, где

E - предикат срабатывания, $E(\beta^1(I(p)), \beta^0(O(p)))$; U - предикат перезапуска, $U(\beta^1(I(p)), \beta^0(O(p)))$; H - функция матрицы маршрутизации, H :

$(\beta^1(I(p)), \beta^0(O(p))) \rightarrow (B^{(|I(p)|+1)})^{O(p)+1}$; r - глубина реентерабельности узла, $r \in N_0$.

Значение функции маршрутизации H вычисленное для узла $p \in P$ на i -ом шаге интерпретации ПМС представляет собой прямоугольную матрицу размерности $(|O(p)| + 1) * (|I(p)| + 1)$ и обозначается $H_i(p)$, причем, 0-я строка образует вектор поглощения, значение 1 в элементе которого означает изъятие фишки из маркировки соответствующей дуги при срабатывании узла, 0-й столбец - вектор генерации, определяющий добавление фишки в маркировку соответствующей

дуги. Остальные элементы матрицы определяют маршрутизацию этих фишек при срабатывании узла: значение элемента, равное 1 означает возможность транспортирования фишки с соответствующего входа на выход, при этом выходная фишка может являться комбинацией нескольких входных и формируется по определенным правилам.

Аналогично, в терминах ПМС любая система также может быть представлена в виде иерархии подсистем, на конечном уровне которой присутствуют только примитивы ПМС - узлы и дуги. При этом любой узел сети может быть заменен соответствующей ему подсетью. Однако, иерархическое представление усложняет формальный анализ структуры и динамики сети в целом, т.к. семантика подсети не является эквивалентом семантики узла ПМС. Отношение иерархических подстановок Ω сети G определяется следующим образом:

$$\Omega = (G, G^*, P^*), \text{ где}$$

G^* - множество подстановок сети G , $G^*(G) = \{G_1, G_2, \dots, G_m\}$; P^* - множество замен сети G , $P^*(G) = \{p^*_1, p^*_2, \dots, p^*_m / \forall p^*_i \in P, (p^*_i, G_i) = 1, i \in [1, m]\}$.

Плоской сетью называется такая сеть G , которая не содержит подсетей, т.е. $G^*(G) = \emptyset$, $P^*(G) = \emptyset$. Плоская сеть может быть образована путем исключения или разворачивания подстановок. Процедура преобразования иерархической ПМС в плоскую легко может быть реализована автоматически по приведенным в диссертации выражениям в виде рекурсивного алгоритма по иерархическому подчинению подсетей или циклического алгоритма по уровням иерархии сети. После этого, плоская сеть может быть подвергнута автоматическому структурному и семантическому анализу с целью определения характеристик исследуемой системы.

Множество C классов компонентов сети G определяется как

$$C = (C^P, C^V, C^T, C^G), \text{ где}$$

C^P - конечное непустое множество классов узлов; C^V - конечное непустое множество классов дуг; C^T - конечное непустое множество классов фишек; C^G - конечное непустое множество классов подсетей.

Множества C^P , C^V , C^T и C^G содержат хотя бы по одному классу C^P_0 , C^V_0 , C^T_0 и C^G_0 соответственно, являющимися базовыми (предками) для всех остальных классов узлов, дуг, фишек и подсетей. В иерархической ПМС множество C^P классов вершин и множество C^V классов дуг образуют независимые, но непрерывные пространства наследования, т.е. элементы этих множеств являются общими и одинаково доступными для всех подсетей вне зависимости от взаимного расположения этих подсетей в общей структурной иерархии графа G . Элементы множества C^T - классы фишек - образуют, с точки зрения наследования, иерархическое дерево типов в соответствии со структурной иерархией сети G таким образом, что в каждой подсети определен ровно один класс, предком которого является класс, определенный в сети, охватывающей данную подсеть. Следовательно, класс фишки, определенный на самом верхнем уровне иерархии ПМС является базовым, или предком, для всех остальных, определенных на последующих уровнях. Этим достигается разрешение проблемы преобразования типов фишек при их переходе в подсеть или из нее, т.к. на каждом последующем уровне доопределяется тот набор данных, который необходим в данной подсети, но не используется на высших уровнях иерархии

ПМС.

Класс узла $c^P \in C^P$ определяется как

$$c^P(c^P_B) = (E, U, H, r, \psi^P, \Sigma), \text{ где}$$

c^P_B - базовый для c^P класс, т.е. класс-предок; E - предикат срабатывания; U - предикат перезапуска; H - функция матрицы маршрутизации; r - глубина реентерабельности; ψ^P - частные данные; Σ - синтаксический класс.

Все соответствующие элементы класса c^P наследуются узлом этого класса. При этом элементы r и ψ^P являются данными узла класса c^P . Значение элемента r может быть изменено правилами выполнения ПМС во время интерпретации сети в соответствии с реализуемым алгоритмом поведения сети. В то же время, ψ^P - частные данные, недоступные правилам выполнения ПМС. Частные данные могут определяться для реализации алгоритмов вычисления предикатов E и U и функции H и могут быть модифицированы только при вычислении значений этих элементов. Синтаксический класс Σ является предикатом, предназначенным для контроля правильности синтаксиса элементарной сети, построенной на базе узла соответствующего класса.

Класс дуги определяется как

$$c^V(c^V_B) = (\eta^L, \eta^H), \text{ где}$$

c^V_B - базовый для c^V класс, т.е. класс-предок; η^L - допустимое минимальное количество фишек в маркировке дуги; η^H — допустимое максимальное количество фишек в маркировке дуги.

Класс дуги не имеет частных данных, так как не имеет методов, для которых они были бы доступны, но при объявлении класса дуги также может быть использована возможность наследования элементов данных, определенных в базовом классе. Обязательно определенный класс дуги c^V_0 , являющийся базовым для любого другого класса дуги, представляет собой $c^V_0 = (0, \infty)$. Также определены классы фишек и классы подсетей и, кроме того, формальные процедуры наследования и инкапсуляции.

Механизмы инкапсуляции и наследования ПМС в соответствии с объектно-ориентированным подходом позволяют разработчику определять классы компонентов ПМС в соответствии с предметной областью исследования для выполнения разнообразных специфических функций. В результате построение модели может выполняться в терминах данной предметной области.

Правила R выполнения сети G позволяют доопределять семантику ПМС, внося зависимость поведения отдельных ее компонентов от состояния сети в целом и внешних факторов. Главным образом, правила R реализуют во время интерпретации сети такие механизмы, как:

- учет временного фактора в условных единицах модельного времени, называемых квантами, в масштабе к реальному времени или без такового;
- выполнение любых вычислений, допустимых программной реализацией среды, использующей аппарат ПМС;
- осуществление обмена информацией между интерпретируемой сетью или построенной на ее базе системой и окружающей ее программной и аппаратной средой;
- управление маршрутизацией фишек в сети в зависимости от каких-либо условий, определяемых состоянием сети или другими, в том числе внешними, факторами;

- адаптивное или функциональное изменение временных и прочих параметров интерпретируемой сети;
- реконфигурация интерпретируемой сети или построенной на ее базе системы в рамках, предусмотренных разработчиком.

Правила выполнения сети определяются как

$$R = (A, D, F, S, \psi^R), \text{ где}$$

A - множество функций определения разрешения срабатывания узлов и маршрутизации фишек по входам и выходам; D - множество функций определения длительности срабатывания узла; F - множество функций, выполняющих преобразования атрибутов перемещаемых фишек и другие вычисления при реализации срабатывания узла; S - множество функций сортировки фишек в дугах; ψ^R - набор частных данных правил выполнения сети.

Элементы множеств A , D , F и S являются функциями, зависящими от состояния как входных и выходных дуг, так и глобальных переменных сети, и вычисляемыми в определенные фазы срабатывания узла. Каждому узлу $p \in P$ может быть сопоставлено не более двух элементов $a(p) \in A$ и не более чем по одному $d(p) \in D$ и $f(p) \in F$. Не более чем один элемент $s(v, t) \in S$ сопоставляется дуге $v \in V$. Если какой-либо элемент не сопоставлен узлу или дуге, то подразумевается использование значения или вычисление функции по умолчанию. Все функции при их вычислении могут использовать или изменять значения элементов набора частных данных ψ^R в соответствии с алгоритмом, заложенным разработчиком.

Правила ПМС позволяют описывать, при помощи данного аппарата, системы управления, отличающиеся динамическим изменением структуры и сложным многорежимным алгоритмом функционирования. Предложена формализованная методика представления в терминах ПМС таких систем, а также показана возможность отражать в модели такие явления, как реентерабельность программного кода и реакцию модели на внешние воздействия программно-аппаратной среды.

Также предложены формальные методы оценки таких структурных и динамических свойств ПМС, как ограниченность и безопасность, сохранение, активность и достижимость маркировки, оперирующие непосредственно формальным представлением ПМС. Приведен разработанный метод определения достижимости маркировки ПМС.

Так как применение методов анализа, разработанных для сетей Петри и их расширений, непосредственно к ПМС невозможно, представлены методики трансформации ПМС в сети Петри и Е-сети и рассмотрены условия и способы сокращения полученной в результате трансформации сети. К трансформированной ПМС могут быть применены методы формального анализа, разработанные специально для данных аппаратов и их модификаций. Данные методы в полной мере могут быть реализованы с использованием средств вычислительной техники.

Показано, что структурные и функциональные возможности формального аппарата ПМС позволяют проводить исследование динамических процессов в сложных распределенных иерархических системах, моделировать алгоритмы управления с учетом асинхронности и одновременности выполнения взаимодействующих процессов, осуществлять преобразование информации,

учитывать временные характеристики событий и их параллелизм.

В третьей главе показано, что формальная система ПМС предоставляет аппарат, позволяющий минимальными выразительными средствами с использованием принципов объектно-ориентированного подхода реализовать базовый набор компонентов, который будет достаточен и наиболее эффективен для решения конкретной задачи или при исследованиях в конкретной предметной области. Базовый набор компонентов содержит некоторое множество классов, состоящее, в основном, из классов узлов и подсетей. Разница состоит в том, что узлы решают задачи, которые являются элементарными для данной предметной области на выбранном уровне детализации описания системы, причем реализация функции узла производится посредством единственного срабатывания. Подсети более сложны как функционально, так и структурно, и их свойства зависят в первую очередь от функциональных свойств узлов, составляющих данную подсеть. Поэтому, реализация функции подсети производится путем реализации некоторой последовательности срабатываний, которая может быть бесконечной. Главной задачей является создание базового набора классов узлов, обеспечивающих реализацию всех функций исследуемой системы.

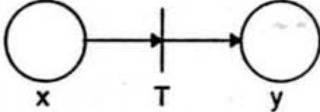
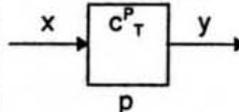
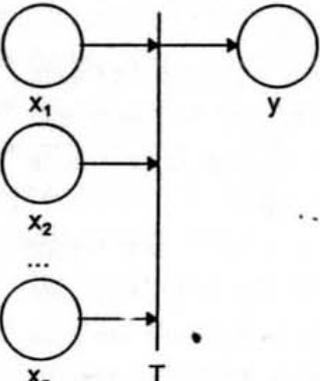
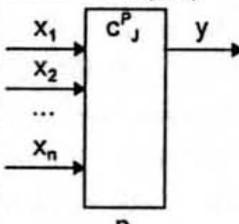
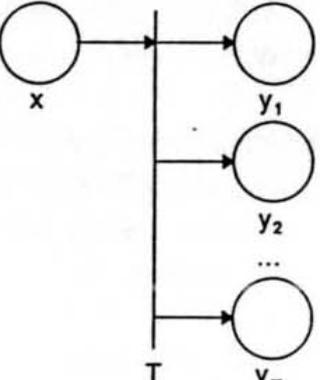
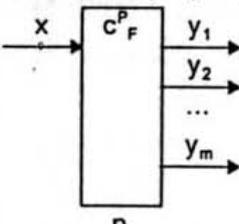
В этой связи рассмотрены способы создания функционально полных базовых наборов компонентов ПМС. Приведены несколько базовых наборов, достаточных для представления любой системы или реализации любого алгоритма, в том числе реализующих машину Тьюринга и неограниченную регистровую машину.

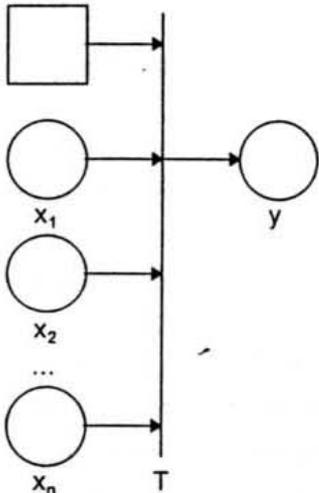
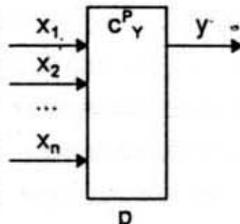
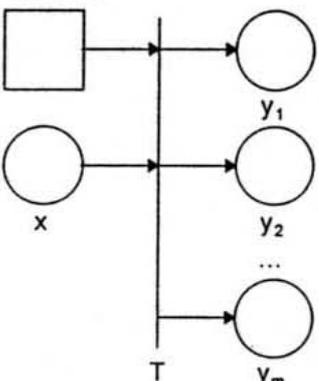
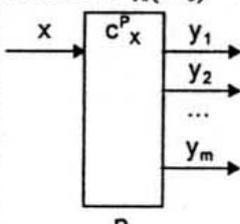
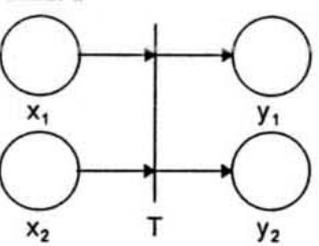
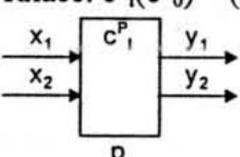
В первом случае доказана возможность трансформации в ПМС E-сетевых структур. E-сети предоставляют набор элементарных схем, в общем, достаточный для описания любого устройства или алгоритма. Возможность этой формальной модели представлять машину Тьюринга доказана, следовательно, возможность представлять E-сети средствами ПМС служат доказательством того факта, что ПМС также эквивалентны машине Тьюринга. В таблице 1 приведены элементарные схемы E-сетей и реализация соответствующих им классов узлов ПМС. В E-сетях обычно выделяют шесть типов элементарных схем, имеющих свои символические обозначения, которые использованы ниже как для элементарных схем E-сетей, так и в качестве индексов в обозначении соответствующих им классов узлов ПМС.

В таблице 1 как x и y обозначены соответственно входные и выходные позиции (в случае E-сетей) или дуги (в случае ПМС). При этом, для элементарной схемы E-сетей приведены в указанном порядке: тип, графическое изображение и условие запуска в виде выражения $s(T)$, в котором $b(x)$ обозначает, что позиция x содержит фишку, а $b(y)$ - позиция y пуста или, если y является позицией-очередью, всегда истинно.

Для классов узлов ПМС в данной таблице приведены: определение класса, вариант графического изображения и определение элементов класса. Те элементы определяемого класса, которые наследуются от базового класса c^p_0 , не присутствуют в определении данного класса. Для придания объектам класса узла определенного поведения необходимо переопределение хотя бы таких элементов, как предикат срабатывания E , функция матрицы маршрутизации H и глубина реентерабельности g .

Элементарные схемы E-сетей и эквивалентные им классы ПМС

Элементарная схема E-сети	Класс узла ПМС
<p style="text-align: center;">1</p> <p>Тип: T</p>  <p>$c(T) = b(x) \wedge \overline{b(y)}$</p>	<p style="text-align: center;">2</p> <p>Класс: $c_T^P(c_0^P) = (E, H, r, \Sigma)$</p>  <p>$E = \beta^I(x) \wedge \beta^O(y)$</p> $H = \begin{vmatrix} 1 & 1 \\ 1 & 1 \end{vmatrix}$ <p>$r = 1$</p> $\Sigma = (I(p) = 1) \wedge (O(p) = 1)$
<p>Тип: J</p>  <p>$c(T) = b(x_1) \wedge b(x_2) \wedge \dots \wedge b(x_n) \wedge \overline{b(y)}$</p>	<p>Класс: $c_J^P(c_0^P) = (E, H, r, \Sigma)$</p>  <p>$E = \beta^I(x_1) \wedge \beta^I(x_2) \wedge \dots \wedge \beta^I(x_n) \wedge \beta^O(y)$</p> $H = \begin{vmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 1 & 1 & \dots & 1 \end{vmatrix}$ <p>$r = 1$</p> $\Sigma = (I(p) > 1) \wedge (O(p) = 1)$
<p>Тип: F</p>  <p>$c(T) = b(x) \wedge \overline{b(y_1)} \wedge \overline{b(y_2)} \wedge \dots \wedge \overline{b(y_m)}$</p>	<p>Класс: $c_F^P(c_0^P) = (E, H, r, \Sigma)$</p>  <p>$E = \beta^I(x) \wedge \beta^O(y_1) \wedge \beta^O(y_2) \wedge \dots \wedge \beta^O(y_m)$</p> $H = \begin{vmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ \dots & \dots \\ 1 & 1 \end{vmatrix}$ <p>$r = 1$</p> $\Sigma = (I(p) = 1) \wedge (O(p) > 1)$

1	2
<p>Тип: Y</p>  <p>$c(T) = b(x_i) \wedge \overline{b(y)}, i = r(T)$</p>	<p>Класс: $c^p_Y(c^p_0) = (E, H, r, \Sigma)$</p>  <p>$E = (\beta^1(x_1) \vee \beta^1(x_2) \vee \dots \vee \beta^1(x_n)) \wedge \beta^0(y)$</p> $H = \left\ \begin{array}{cccc} 1 & (a^1=1) & (a^1=2) & \dots & (a^1=n) \\ 1 & 1 & 1 & \dots & 1 \end{array} \right\ $ <p>$r = 1$ $\Sigma = (I(p) > 1) \wedge (O(p) = 1)$</p>
<p>Тип: X</p>  <p>$c(T) = b(x) \wedge \overline{b(y_i)}, i = r(T)$</p>	<p>Класс: $c^p_X(c^p_0) = (E, H, r, \Sigma)$</p>  <p>$E = \beta^1(x) \wedge (\beta^0(y_1) \vee \beta^0(y_2) \vee \dots \vee \beta^0(y_m))$</p> $H = \left\ \begin{array}{cc} 1 & 1 \\ (a^0=1) & 1 \\ (a^0=2) & 1 \\ \dots & \dots \\ (a^0=m) & 1 \end{array} \right\ $ <p>$r = 1$ $\Sigma = (I(p) = 1) \wedge (O(p) > 1)$</p>
<p>Тип: I</p>  <p>$c_1(T) = b(x_1) \wedge \overline{b(x_2)} \wedge \overline{b(y_1)}$ $c_2(T) = b(x_2) \wedge \overline{b(y_2)}$</p>	<p>Класс: $c^p_I(c^p_0) = (E, H, U, r, \Sigma)$</p>  <p>$E = \beta^1(x_1) \wedge \beta^0(y_1)$</p> $H = \left\ \begin{array}{cc} 1 & 1 \\ 1 & 1 \end{array} \right\ $ <p>$U = \beta^1(x_1) \wedge \beta^1(x_2) \wedge \beta^0(y_1) \wedge \beta^0(y_2)$ $r = 1$ $\Sigma = (I(p) = 2) \wedge (O(p) = 2)$</p>

При вычислении матрицы маршрутизации используются вычисленные ранее значения функций ограничения срабатывания и маршрутизации a^1 и a^0 . Так как в E-сетях срабатывание перехода является его особым состоянием, то одновременно, в отличие от ПМС, не может существовать более одного срабатывания на одном переходе. Вследствие этого, у всех определенных в табл.1 классов глубина реентерабельности γ установлена равной 1, что позволяет одновременно существовать не более чем одному срабатыванию на одном узле.

Способность ПМС представлять неограниченную регистровую машину доказана различными способами, в том числе при помощи базового набора компонентов, реализующего требуемые функции исключительно средствами структуры, что неосуществимо для большинства расширений сетей Петри, но возможно в средствах ПМС благодаря гибкой многоуровневой организации их семантических свойств. Способ, не требующий реализации правил ПМС, заключается в совмещении в одном узле функций декремента и проверки на нуль. В этом случае ветвление в зависимости от условия производится исключительно средствами структуры в результате реализации срабатывания на узле класса c_B^P :

$$c_B^P(c^P_0) = (E, H, r, \Sigma), \text{ где}$$

$$E = (\beta^1(x_1) \wedge \beta^1(x_2) \wedge \beta^0(y_1)) \vee (\beta^1(x_2) \wedge \beta^0(y_2));$$

$$H = \begin{vmatrix} 1 & (a^1 = 1) & 1 \\ (a^1 = 1) & 1 & 1 \\ (a^1 = 2) & 0 & 1 \end{vmatrix}$$

$$\gamma = \infty;$$

$$\Sigma = (\|I(p)\| = 2) \wedge (\|O(p)\| = 2).$$

Операция инкремента выполняется узлом другого класса, например, рассмотренного выше класса c_F^P , имеющего два выхода, к одному из которых подсоединена дуга, представляющая в модели регистр, а другой выход используется для транспортирования фишки к узлу, соответствующему следующему выполняемому оператору алгоритма. Возможны и другие варианты реализации данных операций, например, посредством всего лишь одного класса, который соединяет в себе функции обоих рассмотренных классов c_B^P и c_F^P и, следовательно, образует функционально полный набор компонентов.

Рассмотренные примеры иллюстрируют адаптируемость формального аппарата ПМС к конкретной предметной области, в данном случае, к терминам неограниченной регистровой машины. Кроме того, очевидное преимущество такого задания поведения компонентов ПМС заключается в том, что существует возможность создавать более мелкие (узлы), чем в других аппаратах (подсети, страницы), структурные единицы, выполняющие требуемые функции не последовательностью срабатываний, а только за одно срабатывание, хотя сам аппарат предоставляет возможность описывать иерархические структуры, в том числе с многократным вхождением подсетей одного класса на различных уровнях иерархии.

Также доказано, что формальный аппарат ПМС позволяет реализовать такие операции управления процессами, как разделение и слияние процессов, выбор одного из некоторого множества процессов для дальнейшего исполнения, выбор алгоритма (или средств) дальнейшего выполнения процесса, синхронизация параллельно протекающих процессов, взаимоисключение процессов при доступе к ресурсам. Генерация и уничтожение процессов являются, как правило, частными случаями их разделения и слияния, так как любой процесс в системе либо является ее неотъемлемой частью, либо возникает при выполнении определенных условий, т.е. вследствие некоторых событий или их комбинаций, связанных с развитием процессов в системе, и уничтожается после реализации задачи.

В ПМС любой процесс может быть представлен фишкой. При этом, ее принадлежность маркировке какой-либо дуги определяет состояние соответствующего процесса, а узлы ПМС реализуют операции управления процессами. Операции, производимые самим процессом, могут быть реализованы, при необходимости, в правилах R функциями преобразования F , причем, операндами этих функций являются данные фишек ψ , представляющие переменные процесса.

В ПМС возможны различные варианты реализации одной и той же операции. Несмотря на то, что все варианты решают одну и ту же задачу, это достигается разными средствами, дающими различные возможности разработчику и требующие неодинакового количества компонентов для реализации, которая может быть выполнена наиболее эффективно для выбранного уровня детализации общей задачи и требуемых результатов исследования.

Предложена формализованная методика построения схемы ПМС по диаграмме состояний процесса и реализуемому им алгоритму. Диаграмма состояний представляет собой ориентированный граф, множество вершин которого соответствует множеству состояний системы или процесса, а множество направленных дуг - возможным переходам из одного состояния в другое при наличии некоторого условия. В полученной схеме ПМС каждому состоянию и каждому переходу соответствует дуга, а логика переходов из одного состояния в другое реализуется узлами и связанными с ними правилами ограничения срабатывания и маршрутизации.

Следует отметить, что данная схема отличается свойством строгого сохранения маркировки и может использоваться для моделирования системы с фиксированным количеством однотипных процессов. Для этого необходимо, чтобы глубина реентерабельности узлов была не меньше количества процессов в системе или, в общем случае, неограниченна. Если количество процессов не постоянно и они не являются однотипными, т.е. образуют несколько групп с одинаковыми диаграммами состояний, то результирующая схема ПМС может быть получена путем комбинации нескольких обособленных схем, а создание и уничтожение процессов представляется узлами в разрывах дуг тех переходов, во время которых процессы создаются и уничтожаются.

Схема, построенная по диаграмме состояний процесса, очевидным образом отражает реализуемый им алгоритм и может быть подвергнута дальнейшему уточнению (декомпозиции) до выбранного уровня представления

описываемой системы, в том числе и в комбинации со схемами, рассмотренными ранее, что дает в результате представление различных аспектов структуры и поведения моделируемой системы. Полученная модель в единой системе терминов реализует управление процессами, ресурсами и потоками данных, а также алгоритмов вычислений и обработки информации.

В четвертой главе на основе существующих тенденций развития программных средств автоматизированного проектирования и быстрой разработки приложений раскрываются принципы организации и функционирования системы имитационного моделирования, решающей вопросы исследования систем распределенной обработки информации. Сформулированы требования к модельным исследованиям систем управления и на их основании сделан вывод о необходимости создания объектно-ориентированной среды имитационного моделирования, основанной на формальном аппарате ПМС и обеспечивающей инструментальную поддержку комплекса задач, связанных с исследованием сложных систем управления методом имитационного моделирования.

Предложена концепция построения среды имитационного моделирования с использованием формального аппарата ПМС, обеспечивающей задание структуры и правил модели, определение переменных и результатов моделирования, создание наборов проблемно-ориентированных компонентов ПМС, эффективную отладку моделей, анализ и визуализацию результатов модельного эксперимента. Определены типы, формы представления и порядок обработки статистической информации, получаемой средствами среды моделирования в результате проведения модельного эксперимента, необходимой и достаточной для проведения исследования всех событий и явлений, происходящих в моделируемой системе с целью выявления зависимости свершения конкретных событий от условий, складывающихся в процессе интерпретации модели.

Создана среда имитационного моделирования, использующая для описания моделируемой системы управления формальный аппарат ПМС и обеспечивающая создание, отладку, модификацию и анализ модели, сбор и обработку статистической информации, получаемой в ходе модельного эксперимента. В разработанной среде имитационного моделирования можно выделить несколько типов статистической информации, используемой для анализа результатов моделирования:

1. Исходные данные, представляющие собой информацию о сценарии модельного эксперимента, необходимую для интерпретации данных, полученных в результате реализации этого эксперимента.
2. Первичная статистическая информация, накапливаемая во время интерпретации модели с помощью специальных средств машины ПМС и представляющая собой, в общем случае, подробный протокол интерпретации сети, позволяющий восстановить порядок возникновения событий и состояние модели на любом шаге интерпретации.
3. Информация, отражающая состояние определенных, указываемых пользователем, событий или условий для их свершения по факту их выполнения в модели. Эта информация фиксируется в определенных векторах, называемых векторами значений, и некоторым образом характеризует моделируемую

систему.

4. Условия завершения моделирования, представляющие собой показатели, характеризующие состояние модели на момент окончания процесса интерпретации и причины его окончания.

5. Вторичная статистическая информация, представляющая собой характеристики различных выборок данных из информации предыдущих типов по таким признакам, как номер интерпретации, группа компонентов, интервал времени, доверительная вероятность результатов и т.д., а также корреляционные моменты этих выборок.

6. Интегральные оценки, вычисляемые по специальным функциям, зависимостям и методам для определения всевозможных коэффициентов, показателей, характеризующих поведение исследуемой системы в соответствии с данной предметной областью.

Структура разработанной среды имитационного моделирования при помощи ПМС, включающая средства создания, отладки и эксплуатации моделей, представлена на рис. 1. Здесь сплошными стрелками обозначена статическая подчиненность программного кода, т.е. вызовы, реализованные на этапе разработки и компиляции среды. Штриховые стрелки показывают динамическую подчиненность кода, которая возникает во время функционирования среды моделирования. Это необходимо в том

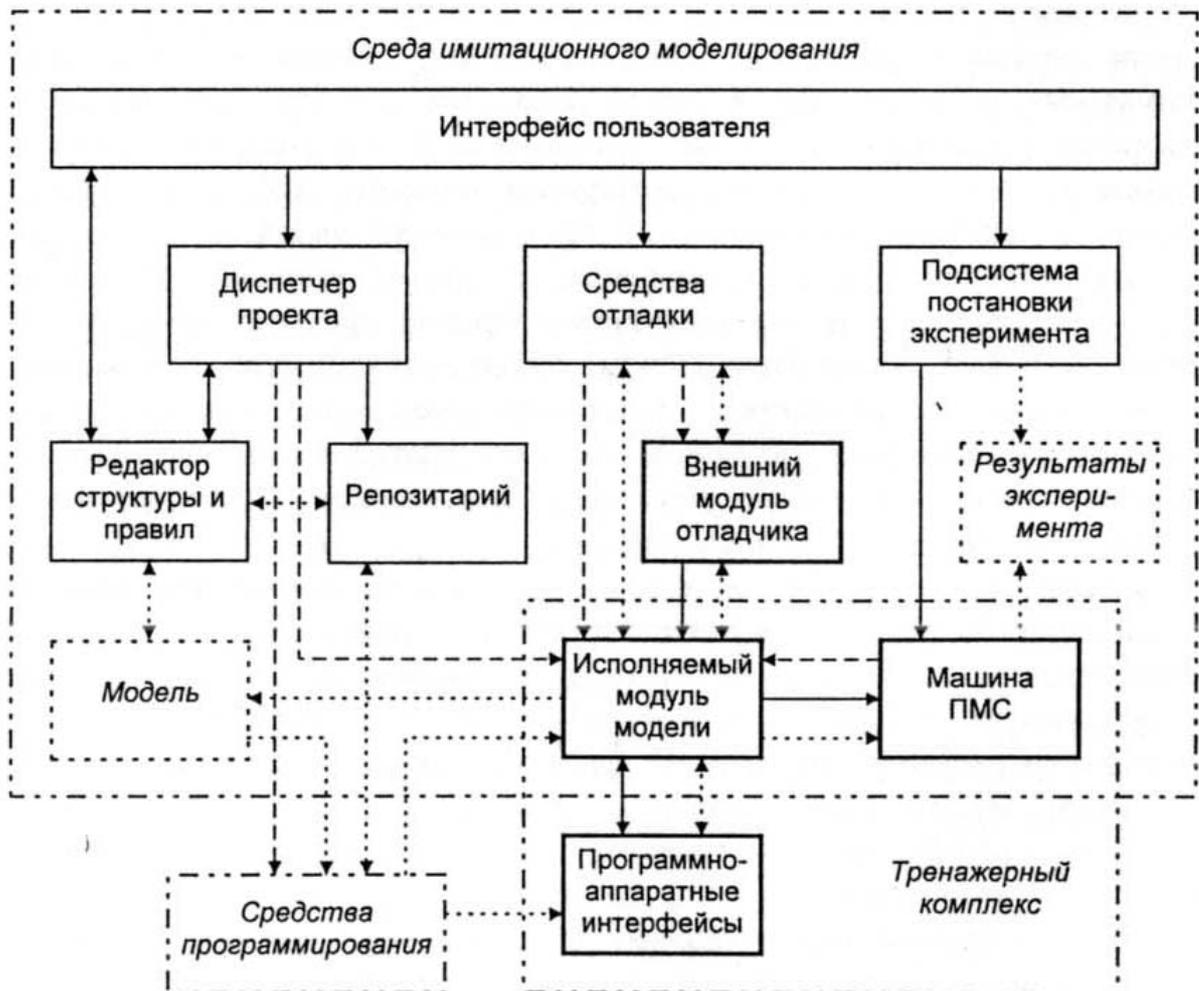


Рисунок 1. Структура среды имитационного моделирования.

случае, если во время компиляции и сборки программы вызываемый в ней код еще не существует. Пунктирные стрелки соответствуют потокам данных. Жирной рамкой выделены те блоки, которые выполнены в виде внешних исполняемых модулей и не являются постоянными (кроме машины ПМС), т.е. могут модифицироваться или заменяться в процессе функционирования среды. Блоки, выделенные пунктирной рамкой, являются структурами данных. Штриховой рамкой обведены группы блоков, образующих функционально полные системы различного назначения. Так, исполняемый модуль модели и машина ПМС присутствуют как в среде моделирования в качестве разрабатываемого и исследуемого объекта, так и в программно-аппаратном тренажерном комплексе в качестве ее ядра. Средства программирования являются самостоятельным программным продуктом, не входящим непосредственно в среду моделирования.

Внутреннее представление разрабатываемой модели является определенным образом организованной структурой данных, хранящей сведения о структуре сети, параметрах ее компонентов, их графического представления, иерархических связей между модулями сети, а также создаваемые разработчиком модели правила в текстовом формате на выбранном языке программирования высокого уровня. Эта информация используется как средствами программирования при создании исполняемого модуля модели, так и самой средой моделирования для представления текущего состояния модели при интерактивной интерпретации сети в режиме отладки.

Исполняемый модуль модели загружается и выполняется под управлением диспетчера проекта и является реализацией структур данных и правил сети, обрабатываемых и вызываемых машиной ПМС по сценарию моделирования, соблюдаемому подсистемой постановки эксперимента. В результате этого процесса информация о ходе моделирования, генерируемая машиной ПМС и подсистемой постановки эксперимента, накапливается и обрабатывается средой моделирования, представляя собой результаты эксперимента. В рамках машины ПМС также реализованы простые в использовании механизмы передачи сообщений произвольного характера, содержание и назначение которых определяются разработчиком модели. Кроме накопления этих сообщений в результатах эксперимента, они могут быть использованы в тренажерных системах при их эксплуатации в целях оперативного мониторинга системы, составления протокола работы, оповещения инструктора или обучаемого о наступлении определенных событий.

В режиме отладки модели информация, генерируемая исполняемым модулем модели, используется для отображения состояния сети, однако, стандартный способ представления этой информации может не позволять увидеть все аспекты этого процесса. Средства отладки позволяют не только наблюдать за поведением компонентов, но и изменять их состояние при необходимости. Однако, реализация в рамках среды моделирования механизма доступа к данным, определенным разработчиком в модели, является трудоемким процессом и приведет в результате к ограничению его возможностей при создании языковых конструкций. Решению этой проблемы служит внешний модуль отладчика, который при необходимости создается и загружается после компиляции модели. Данный модуль осуществляет просмотр и модификацию

значений переменных моделирования по усмотрению разработчика и, как правило, является уникальным для каждой модели. Достоинство такого подхода заключается в том, что внешний модуль отладчика может быть заменен без прерывания процесса интерпретации, причем в этом модуле может производиться любая обработка информации.

Редакторы структуры и правил представляют очень тесно связанные специализированный графический редактор и редактор программного кода. Предполагается, что редакторы также позволяют наглядно представлять иерархическую структуру модели, осуществлять навигацию по ней, отображать состояние модели во время отладки и подавать соответствующие команды и др., т.е. представляют собой основной инструмент разработчика. Кроме того, для адекватного представления и использования компонентов редакторы имеют тесную связь и с репозитарием, осуществляющим хранение библиотек предметно-ориентированных компонентов в виде совокупности структуры, правил и графического представления. Репозитарий может пополняться посредством редакторов структуры и правил.

Такой подход позволяет исследователю путем прямого выбора графического изображения интересующего компонента графа ПМС активизировать редактор кода с целью задания правил, связанных с этим компонентом. Фактически, эффективность разработки и исследования модели может быть существенно повышена за счет наличия множества окон редакторов, отображающих одновременно различные фрагменты сети и различные подсети. При данной конфигурации редакторов исследователь в рамках единой среды имеет возможность контролировать состояние различных компонентов модели, что особенно важно на этапе отладки. Однако, данный подход имеет и такой недостаток, как невозможность оценить получаемый исходный код в полном объеме, но для специалиста предметной области это вряд ли существенно, так как этот код содержит значительные фрагменты, позволяющие представить его как единый модуль с соблюдением всех синтаксических и структурных правил, а также согласования с компонентами среды моделирования. С другой стороны, представление исходного кода в виде набора фрагментов, лишенных служебной информации и управляемых средой моделирования, является удобным для конечного пользователя.

Программно-аппаратные интерфейсы, присутствующие только в составе тренажерного комплекса, обеспечивают требуемое представление и управление последним как самостоятельным программным продуктом, взаимодействие с операционной системой, системным, прикладным программным обеспечением, стандартным и нестандартным аппаратным обеспечением, специальным и технологическим оборудованием (например, пульт оператора) и т.д. Разработка и отладка этого модуля производится средствами программирования без участия среды моделирования, а его взаимодействие с исполняемым модулем модели производится в обоих направлениях посредством вызова интерфейсных функций и модификации интерфейсных переменных правилами модели под воздействием событий сети и вызовом методов объектов модели и модификации модельных переменных интерфейсными функциями под воздействием внешних событий.

На приведенной структурной схеме не отражены второстепенные взаимосвязи между компонентами среды моделирования, как и связь с

программным интерфейсом операционной системы, которая является платформозависимой. Также не показаны некоторые несущественные компоненты среды, относящиеся, скорее, к деталям реализации.

Набор возможных средств программирования предусматривается заранее, так как данные средства определяют язык программирования, предоставляемый исследователю, и, следовательно, формат исходного кода модуля модели, генерируемого средой моделирования. Информация об используемых средствах представлена в виде файлов шаблонов и конфигураций, что обеспечивает относительно простую адаптацию среды моделирования к требованиям исследователей. В существующем варианте среда имитационного моделирования рассчитана на совместную работу со средой быстрой разработки приложений Borland C++ Builder, имеющей наиболее развитую спецификацию объектно-ориентированного языка программирования, что позволяет предоставить пользователю наиболее удобные в синтаксисе и в применении объекты реализации компонентов.

Реализована возможность экспорта исходного кода модуля модели и компонентов репозитория в виде проекта в формате Borland C++ Builder, что дает возможность использовать его в качестве алгоритмического ядра разрабатываемых программных продуктов. Конвертеры экспорта и импорта файлов выполнены в виде исполняемых модулей, что придает среде моделирования дополнительную гибкость и свойство быстрой адаптации к постоянно меняющимся условиям рынка программных средств.

Реализация данной среды для работы под управлением многозадачной операционной системы Windows позволила выполнять интерпретацию сети в отдельном программном процессе, что дает большую эффективность моделирования вследствие независимости исполняемой модели от интерфейсной части, перегруженной ресурсоемкими графическими операциями. Кроме того, все исполняемые модули выполнены в формате Dynamic Link Libraries (DLL), что упрощает их создание, отладку и эксплуатацию и, следовательно, дальнейшее развитие среды имитационного моделирования.

Реализация открытого программного интерфейса модели, представленной в виде исполняемого модуля, позволяет осуществлять эффективное взаимодействие последней с окружающей программно-аппаратной средой и дает возможность использования разрабатываемой модели в качестве алгоритмического ядра при построении программно-аппаратных комплексов. Суть открытого программного интерфейса заключается в спецификации, позволяющей организовывать взаимодействия широкого спектра программных модулей, разработанных при помощи различных средств и языков программирования. Единственное требование к этим компонентам заключается в соответствии разработанной спецификации, которая включает в себя типы данных, имена доступных извне функций, механизмы передачи сообщений и перечень возможных сообщений. Программная модель, как объект, связана с окружающей средой посредством следующих информационных потоков:

- управление процессом моделирования;
- воздействие окружающей среды на модель, т.е. влияние внешних процессов на процессы, протекающие внутри модели;
- воздействие модели на окружающую среду вследствие происходящих в модели

событий.

В исполняемом модуле модели реализованы соответствующие группы средств, обеспечивающие описанные взаимодействия. Благодаря открытому программному интерфейсу, модель, разработанная в данной среде имитационного моделирования, является самостоятельным программным модулем, который может быть использован в разработке других программно-аппаратных систем.

В заключении приведены выводы по диссертации, сформулированы полученные научные и практические результаты.

В приложении к диссертации приведены фрагменты исходных текстов разработанной программной среды имитационного моделирования, иллюстрирующие реализацию основных алгоритмов формального аппарата ПМС.

Основные результаты работы

В настоящей диссертации предложен формальный аппарат предикатно-матричных сетей для решения задач описания и анализа сложных систем управления. На основе теории предикатно-матричных сетей - теоретического результата диссертации, разработаны новые методики и средства имитационного моделирования, позволяющие повысить эффективность решения задач проектирования и анализа систем управления.

Результаты диссертационной работы заключаются в следующем:

1. На основе анализа тенденций развития, существующих проблем и методов анализа современных систем управления обоснована необходимость поиска новых подходов к построению и анализу имитационных моделей систем данного класса.
2. Для адекватного модельного представления систем управления, характеризующихся наличием и разнообразием одновременно протекающих в них процессов, впервые предложен класс предикатно-матричных сетей, динамика которых описывается совокупностью предикатов, матриц и правил, а также разработаны формальные методы исследования свойств предикатно-матричных сетей.
3. Разработан алгоритм интерпретации предикатно-матричных сетей, позволяющий получить требуемые динамические характеристики исследуемой системы и хорошо реализуемый средствами вычислительной техники.
4. Доказана алгоритмическая и функциональная полнота предикатно-матричных сетей как формального аппарата и предложен формализованный метод построения модели системы управления по ее описаниям в различных аспектах организации и функционирования.
5. Рассмотрены вопросы взаимодействия модели, представленной в виде исполняемого файла, с внешней программно-аппаратной средой посредством открытого программного интерфейса, спецификация которого предусматривает управление процессом интерпретации модели, ввод в модель информации и генерацию сообщений о происходящих в модели событиях, что позволяет применять разрабатываемые модели как самостоятельные модули в качестве алгоритмического ядра программно-аппаратных комплексов.
6. Созданы программные средства имитационного моделирования, позволяющие при помощи формального аппарата предикатно-матричных сетей производить

описание и анализ сложных систем управления, включая этапы создания, отладки, эксплуатации моделей, накопления и обработки статистической информации, получаемой в результате модельного эксперимента.

Результаты диссертации могут быть использованы не только при разработке и анализе систем управления, но и в задачах исследования систем распределенной обработки информации. Теоретические результаты, связанные с исследованием свойств сетевых моделей, могут быть применены при решении других задач описания и реализации алгоритмов и взаимодействия процессов.

Перечень работ по теме диссертации

1. G.P. Tsapko, P.V. Grishmanovsky. The Imitation E-Net Modeling Toolkit for Complicated Systems. // Abstracts KORUS'97. The First Russian-Korean International Symposium on Science and Technology. - September 29 - October 3, 1997, University of Ulsan, Ulsan, Republic of Korea.
2. G.P. Tsapko, P.V. Grishmanovsky. The New Approach to Organization of E-Net Modeling Toolkit. // Abstracts KORUS'97. The First Russian-Korean International Symposium on Science and Technology. - September 29 - October 3, 1997, University of Ulsan, Ulsan, Republic of Korea.
3. G.P. Tsapko, P.V. Grishmanovsky. The Imitation E-Net Modeling Toolkit for Complicated Systems. // Proceedings KORUS'97. The First Russian-Korean International Symposium on Science and Technology. - September 29 - October 3, 1997, University of Ulsan, Ulsan, Republic of Korea.
4. Запевалов А.В., Запевалова Л.Ю., Гришмановский П.В. Диалоговое моделирование в научно-исследовательской работе. // Сборник научных трудов Сургутского государственного университета. Выпуск 4. Естественные науки. - Сургут: Сургутский государственный университет, 1998.
5. Плюснин И.И., Гришмановский П.В. Оценка эффективности локальной терминальной сети посредством имитационного моделирования. // Сборник научных трудов Сургутского государственного университета. Выпуск 4. Естественные науки. - Сургут: Сургутский государственный университет, 1998.
6. G.P. Tsapko, P.V. Grishmanovsky. Object-Oriented Approach in E-Net Simulation. // Abstracts KORUS'99. The Third Russian-Korean International Symposium on Science and Technology. - June 22-25, 1999, Novosibirsk State Technical University, Novosibirsk, Russia.
7. G.P. Tsapko, P.V. Grishmanovsky. The E-Net as Concurrent Programming Language. // Abstracts KORUS'99. The Third Russian-Korean International Symposium on Science and Technology. - June 22-25, 1999, Novosibirsk State Technical University, Novosibirsk, Russia.
8. G.P. Tsapko, P.V. Grishmanovsky, I.N. Danilenko. Object-Oriented Approach in E-Net Simulation. // Proceedings KORUS'99. The Third Russian-Korean International Symposium on Science and Technology. - June 22-25, 1999, Novosibirsk State Technical University, Novosibirsk, Russia.
9. Гришмановский П.В., Балаканов А.Р. Система E-сетевого имитационного моделирования. // Радиотехнические и информационные системы и устройства. Тезисы докладов региональной научно-технической конференции студентов и молодых специалистов. Часть 1. Автоматизированные системы обработки информации, управления и проектирования. - Томск: Томский государственный

университет систем управления и радиоэлектроники, 1999.

10. Гришмановский П.В., Балаканов А.Р. Система E-сетевого имитационного моделирования. // Современные техника и технологии. Тезисы докладов 5-ой областной научно-практической конференции студентов, аспирантов и молодых ученых. - Томск: Томский политехнический университет, 1999.

11. Цапко Г.П., Гришмановский П.В. Поддержка учебного процесса средствами имитационного моделирования. // Сборник трудов 7-ой Международной научно-методической конференции «Новые информационные технологии в университетском образовании». - Новосибирск: Институт дискретной математики и информатики, 2000.

12. G.P. Tsapko, P.V. Grishmanovsky. Simulation Tools For Development Of Distributed Software System Kernel // Abstracts KORUS'2000. The Fourth Russian-Korean International Symposium on Science and Technology. - 2000, University of Ulsan, Ulsan, Republic of Korea.

13. G.P. Tsapko, P.V. Grishmanovsky., I.N. Danilenko, A.R. Balakanov. Simulation Models At The Development Of Distributed Data Processing Systems // Proceedings KORUS'2000. The Fourth Russian-Korean International Symposium on Science and Technology. - 2000, University of Ulsan, Ulsan, Republic of Korea.